RE

ransactions

ELECTRONIC COMPUTERS

Volume EC-7 JUNE, 1958 Number 2

Published Quarterly

TABLE OF CONTENTS

CONTRIBUTIONS

SYMPOSIUM ON COMPUTERS IN SIMULATION, DATA REDUCTION, AND CONTROL

CORRESPONDENCE

PUBLISHED BY THE

essional Group on ELECTRONIC COMPUTERS

IRE PROFESSIONAL GROUP ON ELECTRONIC COMPUTERS

The Professional Group on Electronic Computers is an association of IRE members with professional interest in the field of Electronic Computers. All IRE members are eligible for membership, and will receive all Group publications upon payment of a fee of $2.00 per year, 1958.

IRE Transactions® on Electronic Computers

Published by the Institute of Radio Engineers, Inc., for the Professional Group on Electronic Computers at 1 East 79th Street, New York 21, N.Y. Responsibility for the contents rests upon the authors and not upon the IRE, the Group, or its members. Price per copy: IRE-PGEC members, $1.50; IRE members, $2.25, nonmembers, $4.50. Yearly subscriptions rate: nonmembers, $17.00; colleges and public libraries, $12.75. Address requests to The Institute of Radio Engineers, 1 East 79th Street, N.Y. 21, N.Y.

Notice to Authors: Address all papers and editorial correspondence to J. P. Nash, Missile Systems Division, Lockheed Aircraft Corp., 3251 Hanover Street, Palo Alto, Calif. To avoid delay, 3 copies of papers and figures should be submitted, together with the originals of the figures which will be returned on request. All material will be returned if a paper is not accepted.

Nonlinear Transfer Functions with Thyrite*

L. D. KOVACH† AND W. COMLEY†

Summary—Since the publication of a previous paper[1] the uses hyrite[2] in the synthesis of nonlinear transfer functions in analog puting have been greatly expanded. Moreover, new and con-rably improved methods of Thyrite selection and matching have n developed.

The present paper shows how rational exponents can be repre-ted. Functions of the form $y=kx^n$, with $1/\delta \leq n \leq 6$ are readily ained. Circuit configurations generating the functions

$$1-\left(\frac{2x}{\pi}\right)^{1.74} \text{ and } 1-\left[\frac{2}{\pi}\left(x-\frac{\pi}{2}\right)\right]^{1.74}$$

ch provide excellent approximations to the cosine and sine func-s, respectively, are given and an improved quarter-square multi-r using Thyrite squaring is discussed.

A study was also made of the physical factors influencing the formance and accuracy of Thyrite and quantitative experimental a regarding these factors are presented.

Characteristics of Thyrite

THYRITE resistors are characterized by a non-linear behavior in that the current varies as some power of the applied voltage. This behavior may expressed generally by (1), where n varies over a nsiderable range.

$$i = ke^n. \tag{1}$$

e value of n is primarily dependent upon the type of yrite and, to a lesser extent, upon the nonuniformity ind among samples of a given type. Fortunately, the aracteristics of any given sample will be relatively nstant when operated conservatively and within the en power rating.

The most suitable Thyrite types for the generation a given exponent are given below.

Catalog Number	*Exponent Range*	*Maximum Current*
8396839G1	$1 \leq n < 2.5$	5 ma
8396832G1	$2.5 \leq n \leq 3$	10 ma

The behavior of the Thyrite resistor as given in (1) ıst generally be modified in order to render the mate-l useful for computing applications. The basic ex-nent of the nonlinear resistor must be somewhat gher than that actually required for a given applica-on. Resistance is connected in series, thereby tending linearize the characteristic of the combination. This, course, reduces the value of n. The amount of re-stance which must be added will be determined by the ecise value of n required. After proper adjustment, the voltage drop across the series resistance will bear the relationship with the input voltage given in (2).

$$e_o = ke_i^n. \tag{2}$$

In order that the exponent remain constant over a wide range of input voltages the original value of n for the Thyrite should not be significantly higher than the final required value. It is fortunate that the value of n for a given type of Thyrite resistor will vary as much as ± 20 per cent. This makes it possible to select samples particularly suitable for a wide variety of applications.

Factors Affecting Over-All Accuracy

An extensive program of tests was conducted to determine the accuracy obtainable with Thyrite and to define those factors which have significant effect. The basic accuracy of the squaring operation, *i.e.*, the conformity of the Thyrite to the true square law was found to vary between 0.3 per cent and 1.0 per cent of full-scale for most samples. A surprisingly large number yielded accuracies better than 0.5 per cent.

The basic accuracy was determined by means of the system illustrated in Fig. 1. The method is based upon the comparison of the simulated function with the function as generated by the proper number of successive integrations of a constant voltage. In this manner a perfect parabola or cubic may be generated for the purpose of comparison. A typical error curve obtained in this manner is shown in Fig. 2. The importance of obtaining the optimum values of series resistance for a given Thyrite sample is illustrated in Fig. 3 where, for several samples, the series resistance has been plotted against the resulting basic accuracy. An examination of the curves indicates clearly that there is a unique value of series resistance which yields the best result.

It was found that the optimum current range was between 0 and 5 ma. Larger currents gave less accurate results due principally to internal heating. Overloads of 100 per cent will not cause permanent modification in the characteristics of the material, but larger, prolonged overloads may result in such changes as to necessitate a redetermination of the optimum series resistance value. Some means of overload protection, such as amplitude limiting, may be desirable in those cases where problem variable peak voltages are not sufficiently predictable.

The adverse effects due to temperature have been considered separately by means of the circuit of Fig. 4. This arrangement provides the means for applying a constant input voltage and monitoring the output. The importance of restricting the current range is indicated in Fig. 5 which indicates the accuracy for various con-

* Manuscript received by the PGEC, April 26, 1957; revised nuscript received, January 23, 1958.

† Douglas Aircraft Co., Inc., El Segundo, Calif.

[1] L. D. Kovach and W. Comley, "An analog multiplier using yrite," IRE Trans. on Electronic Computers, vol. EC-3, pp. -45; June, 1954.

[2] Thyrite is a registered trademark of the Carboloy Dept. of General Electric Co.

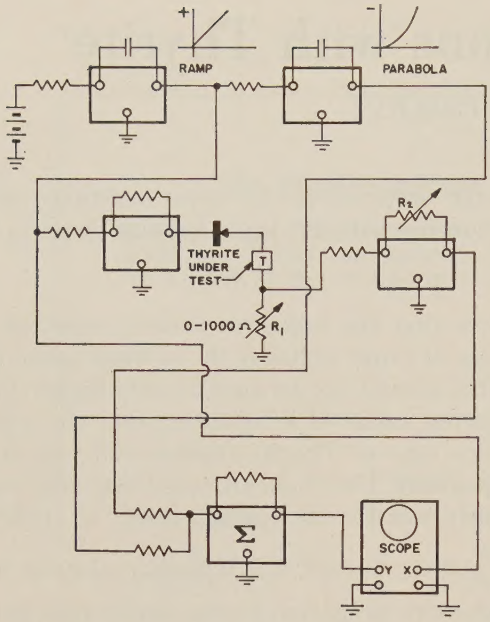


Fig. 1—Circuit for Thyrite selection.

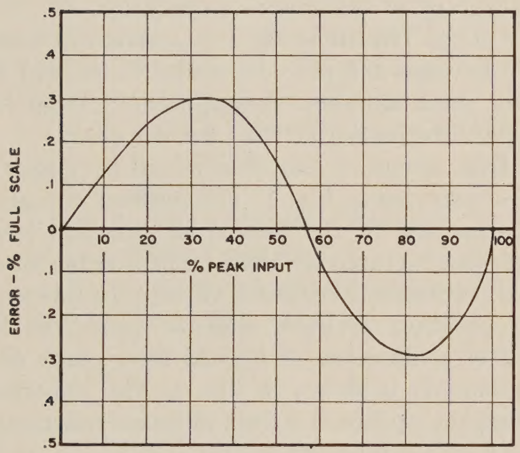


Fig. 2—Typical error curve.

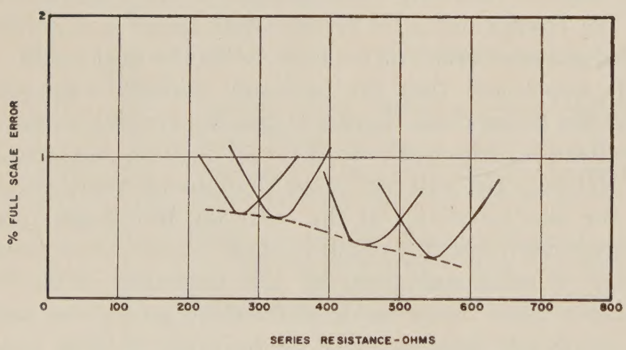


Fig. 3—Squaring error vs series resistance.

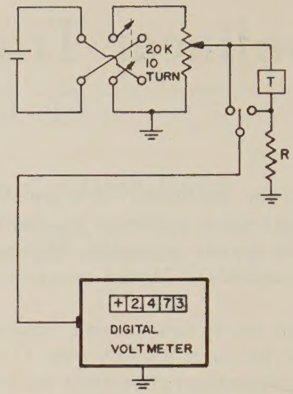


Fig. 4—Circuit for determining heating and semiconductivity effects.

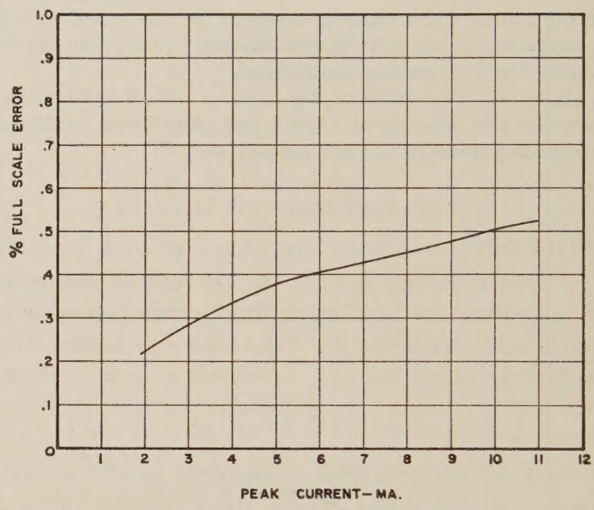


Fig. 5—Error due to internal heating.

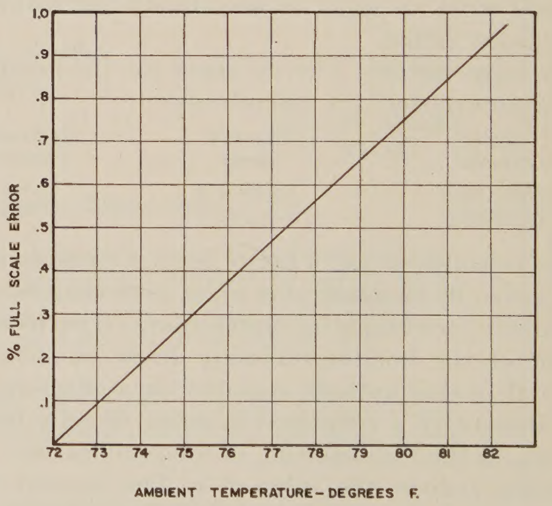


Fig. 6—Error due to ambient temperature variation.

tinuous peak current values. At each point of this curve sufficient time was allowed for the error to stabilize at its maximum value. It was found that the error for the 5 ma peak value was about 0.4 per cent. The continuous application of full current is, of course, a severe condition that will rarely be encountered in operation. In actual practice it is doubtful if the error due to internal heating will ever be as large as 0.2 per cent.

Reduction of the accuracy due to ambient temperature variation is dependent entirely on the environment in which the material is used. For this reason it is advisable to operate the Thyrite at a distance from any heat generating equipment or, if possible, in a constant temperature environment such as provided by the ovens

certain types of analog computers. Fig. 6 shows the l-scale error that will result for variations in the am- nt temperature up to 10°F.

Fortunately, most up-to-date analog installations are conditioned and the room temperature does not vary re than one or two degrees. Moreover, Thyrite is entially insensitive to humidity variations.

Finally, it should be pointed out that Thyrite dis- ys a slightly semiconductive effect. The difference in tput for equal positive or negative inputs may amount as much as 0.3 per cent of full-scale. Assuming that of the adverse effects are additive, it is still apparent at accuracies of 1 per cent of full-scale are not difficult obtain.

In using Thyrite for multiplying by means of the arter-square law two pieces of Thyrite are required. ice the error curves of the two pieces may be entirely fferent, it might appear that increased accuracy can achieved by careful selection. However, it can be own analytically that regardless of the shape of the ror curves the full-scale accuracy of the multiplication ll be the sum of the maximum errors of the individual ces of Thyrite.

It should be pointed out that the results presented in s section apply to the Thyrite having the lower ex- nent. Error evaluation on the type suitable for higher ponents yielded results similar to those previously ntioned, except that a maximum current of 10 ma is rmissible.

Selection and Adjustment of Exponent

The circuit configurations for generating various ictions are given in Table I (pp. 94–96), but first, table Thyrite resistors and series linear resist- ce must be properly chosen and combined in order to ovide the required exponents as indicated in the table. e recommended procedure for this selection and ad- stment has been indicated in Fig. 1. This method is table only for the selection and adjustment of com- iations intended to provide exponents of 2 and 3. For nintegral powers it will become necessary to replace e second integrator in Fig. 1 with a sufficiently ac- rate function generator which has been adjusted to ovide the exact function required. The difference of e two functions is then taken in a summing amplifier e output of which may be observed on a suitable re- rding device. A repetitive setup and oscilloscope will ovide the most rapid and most accurate means for king the necessary adjustments. Adjustment may w be made by varying the Thyrite series resistor, at e same time maintaining the output amplitude. The sired condition will be indicated as the difference of e two functions approaches zero. It should be em- asized that the peak amplitude of the derived function ould be maintained at a level such that the current es not exceed the recommended value for maximum curacy. If variation of series resistance between the its of approximately 300 to 1000 ohms does not result a suitable minimum difference between the two functions, the process should be repeated with other Thyrite samples until the required degree of correspondence between the two functions is indicated. This difference function is the absolute error and may be measured to determine whether or not the required degree of accuracy has been obtained.

Applications

Table I presents in a convenient form the circuit configurations and transfer functions which may be obtained by use of the Thyrite and linear resistance combinations providing the various exponents as determined by the methods outlined in the previous section. Use of these combinations in the input, feedback, and output of operational amplifiers provides a wide range of useful functions. Because of the passive nature of the nonlinear resistance, insertion losses will result. For a typical squaring operation this loss is approximately 20. All or part of this may be compensated for by appropriate gains in the associated amplifiers. It is interesting to note that no Thyrite resistor combination providing an exponent greater than three has been used, although units capable of somewhat higher exponents are available. The excessively high voltages required to place these within their optimum range makes their use impractical in computing applications. It will be noted that in some instances diodes have been used to obtain an absolute value, thereby providing the symmetry required of the even functions.

It should be pointed out that in circuit configurations 7, 8, and 9 of the table the Thyrite is in the input circuit of the amplifier. Due to the variable input impedance in these cases, these circuits should not be supplied from potentiometers or other high impedance devices.

The circuit configurations for the sine and cosine functions use a parabola-like function which has been translated in a suitable manner in order to simulate the desired trigonometric function. The Thyrite resistor combination has been adjusted to provide a power of 1.74. It can be shown that this exponent produces the best least squares fit to the trigonometric functions. Under these conditions the error resulting from the approximation will not exceed 1.35 per cent. The angles which may be resolved, however, will be subject to the restrictions indicated in the table.

An improved quarter-square multiplier circuit is shown in Fig. 7 (p. 97). Balance potentiometers have been added to the squaring circuits in order that the forward resistance of the diodes may be precisely matched, thereby preventing nonsymmetry in the squaring operation. This type of multiplier has several attractive advantages worthy of consideration: 1) high input impedance; 2) no requirement for special equipment; 3) wide-band response; and 4) zero output for zero input, a difficult characteristic to achieve with conventional multiplier schemes. It should be noted that all elements comprising the multiplier are either linear or passive. As a result accuracy is in no way dependent upon active nonlinear devices.

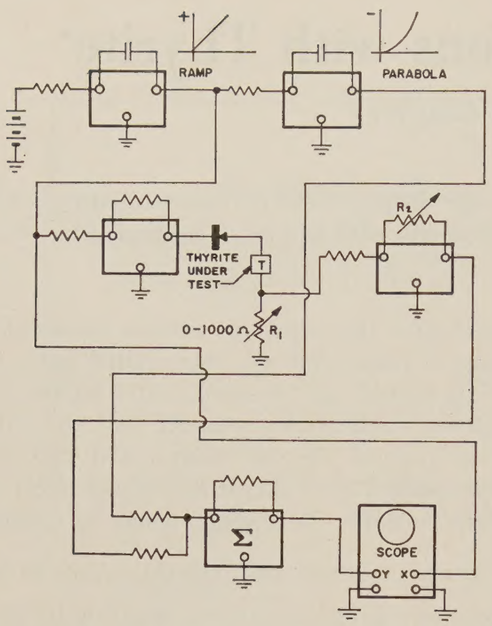


Fig. 1—Circuit for Thyrite selection.

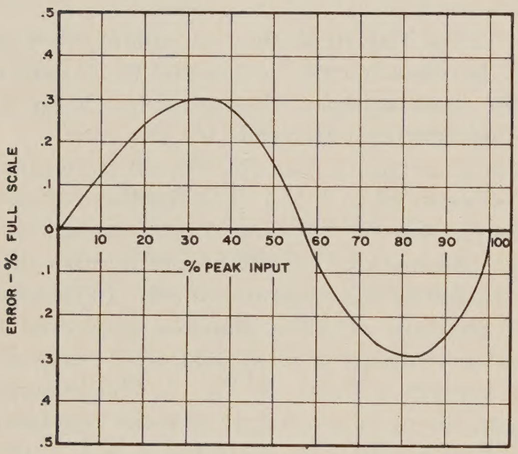


Fig. 2—Typical error curve.

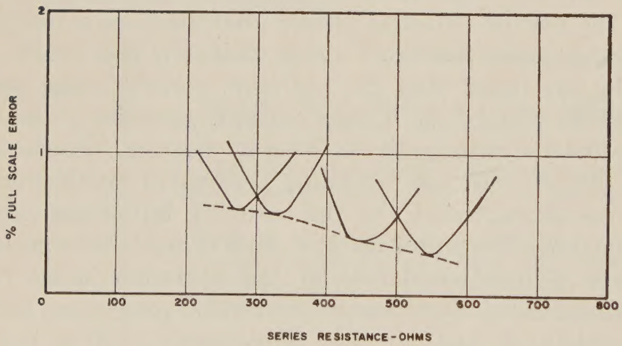


Fig. 3—Squaring error vs series resistance.

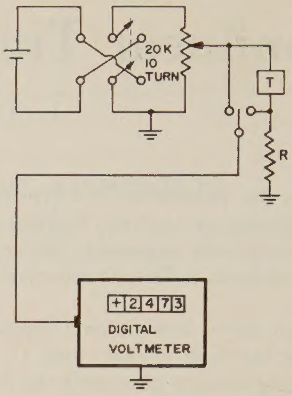


Fig. 4—Circuit for determining heating and semiconductivity effects.

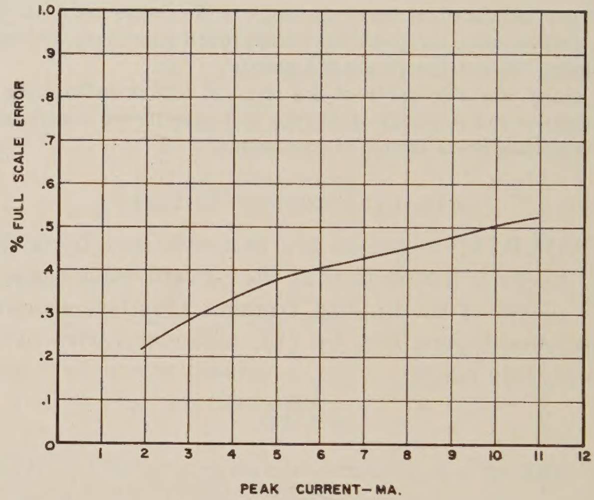


Fig. 5—Error due to internal heating.

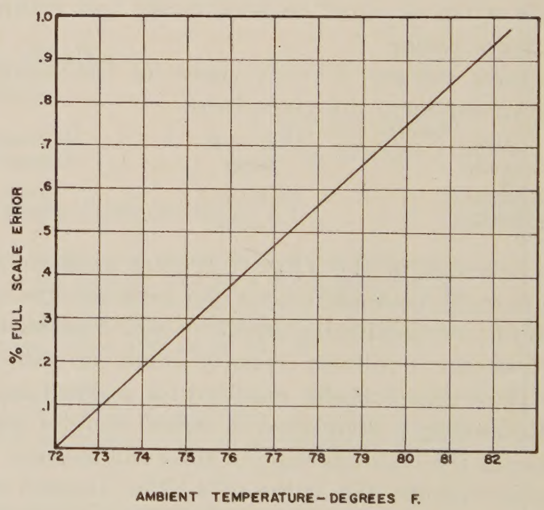


Fig. 6—Error due to ambient temperature variation.

tinuous peak current values. At each point of this curve sufficient time was allowed for the error to stabilize at its maximum value. It was found that the error for the 5 ma peak value was about 0.4 per cent. The continuous application of full current is, of course, a severe condition that will rarely be encountered in operation. In actual practice it is doubtful if the error due to internal heating will ever be as large as 0.2 per cent.

Reduction of the accuracy due to ambient temperature variation is dependent entirely on the environment in which the material is used. For this reason it is advisable to operate the Thyrite at a distance from any heat generating equipment or, if possible, in a constant temperature environment such as provided by the ovens

certain types of analog computers. Fig. 6 shows the l-scale error that will result for variations in the am-nt temperature up to 10°F.

Fortunately, most up-to-date analog installations are conditioned and the room temperature does not vary re than one or two degrees. Moreover, Thyrite is entially insensitive to humidity variations.

Finally, it should be pointed out that Thyrite dis-ys a slightly semiconductive effect. The difference in tput for equal positive or negative inputs may amount as much as 0.3 per cent of full-scale. Assuming that of the adverse effects are additive, it is still apparent it accuracies of 1 per cent of full-scale are not difficult obtain.

In using Thyrite for multiplying by means of the arter-square law two pieces of Thyrite are required. ice the error curves of the two pieces may be entirely ferent, it might appear that increased accuracy can achieved by careful selection. However, it can be own analytically that regardless of the shape of the or curves the full-scale accuracy of the multiplication l be the sum of the maximum errors of the individual eces of Thyrite.

It should be pointed out that the results presented in s section apply to the Thyrite having the lower ex-nent. Error evaluation on the type suitable for higher ponents yielded results similar to those previously ntioned, except that a maximum current of 10 ma is missible.

Selection and Adjustment of Exponent

The circuit configurations for generating various ictions are given in Table I (pp. 94–96), but first, table Thyrite resistors and series linear resist-ce must be properly chosen and combined in order to ovide the required exponents as indicated in the table. e recommended procedure for this selection and ad-tment has been indicated in Fig. 1. This method is table only for the selection and adjustment of com-iations intended to provide exponents of 2 and 3. For nintegral powers it will become necessary to replace second integrator in Fig. 1 with a sufficiently ac-rate function generator which has been adjusted to ovide the exact function required. The difference of two functions is then taken in a summing amplifier output of which may be observed on a suitable re-rding device. A repetitive setup and oscilloscope will ovide the most rapid and most accurate means for king the necessary adjustments. Adjustment may w be made by varying the Thyrite series resistor, at same time maintaining the output amplitude. The sired condition will be indicated as the difference of two functions approaches zero. It should be em-asized that the peak amplitude of the derived function ould be maintained at a level such that the current es not exceed the recommended value for maximum curacy. If variation of series resistance between the its of approximately 300 to 1000 ohms does not result a suitable minimum difference between the two functions, the process should be repeated with other Thyrite samples until the required degree of correspondence between the two functions is indicated. This difference function is the absolute error and may be measured to determine whether or not the required degree of accuracy has been obtained.

Applications

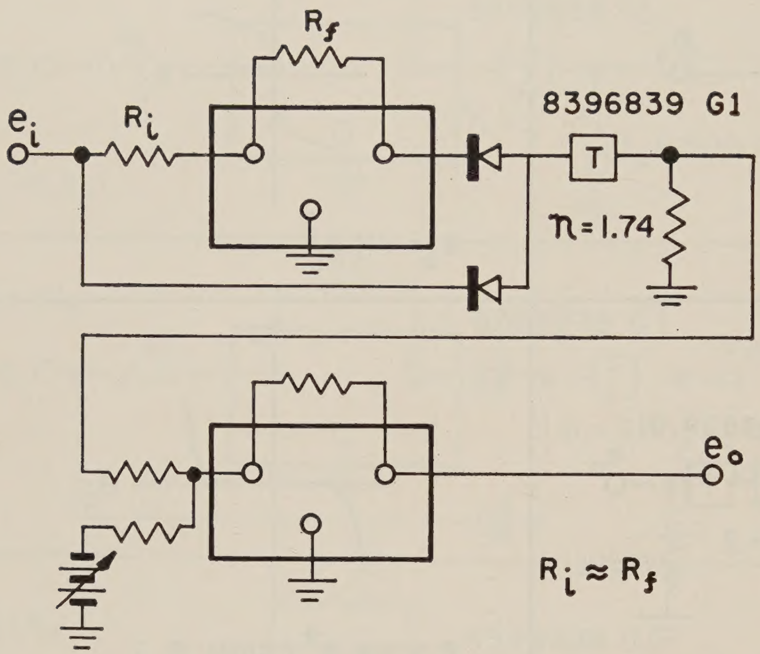
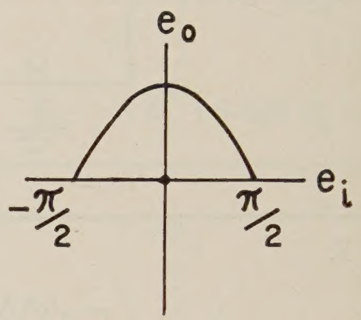
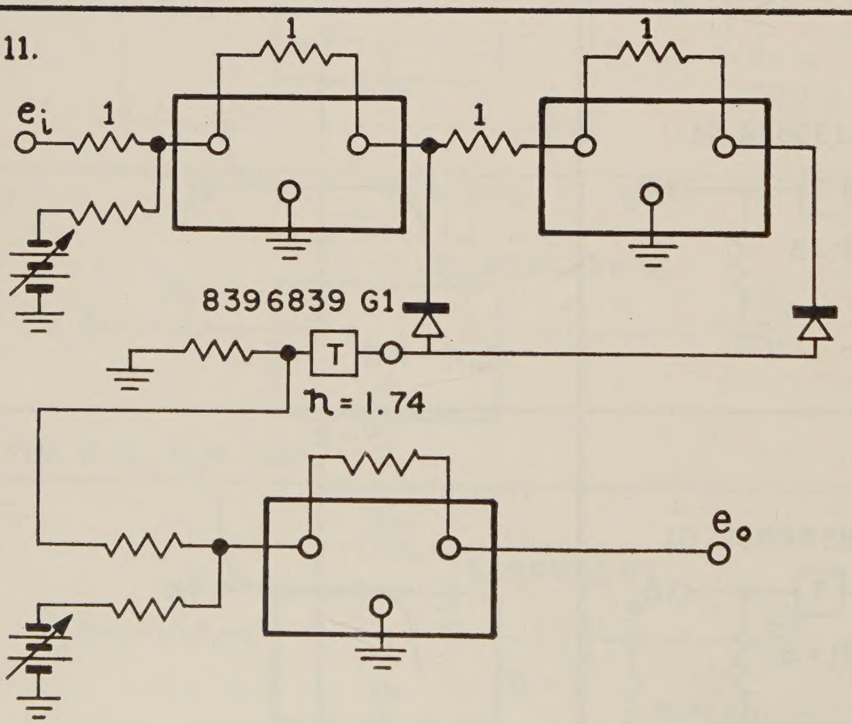
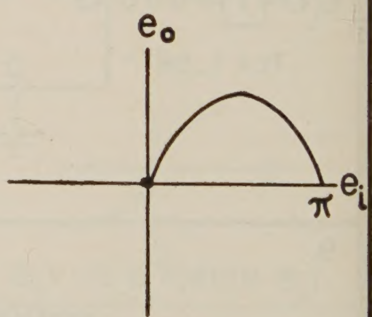
Table I presents in a convenient form the circuit configurations and transfer functions which may be obtained by use of the Thyrite and linear resistance combinations providing the various exponents as determined by the methods outlined in the previous section. Use of these combinations in the input, feedback, and output of operational amplifiers provides a wide range of useful functions. Because of the passive nature of the nonlinear resistance, insertion losses will result. For a typical squaring operation this loss is approximately 20. All or part of this may be compensated for by appropriate gains in the associated amplifiers. It is interesting to note that no Thyrite resistor combination providing an exponent greater than three has been used, although units capable of somewhat higher exponents are available. The excessively high voltages required to place these within their optimum range makes their use impractical in computing applications. It will be noted that in some instances diodes have been used to obtain an absolute value, thereby providing the symmetry required of the even functions.

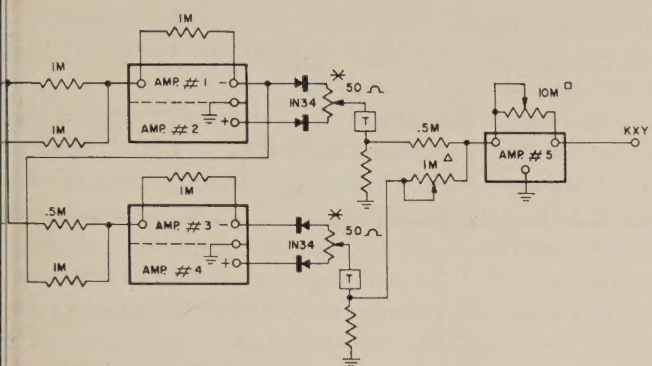
It should be pointed out that in circuit configurations 7, 8, and 9 of the table the Thyrite is in the input circuit of the amplifier. Due to the variable input impedance in these cases, these circuits should not be supplied from potentiometers or other high impedance devices.

The circuit configurations for the sine and cosine functions use a parabola-like function which has been translated in a suitable manner in order to simulate the desired trigonometric function. The Thyrite resistor combination has been adjusted to provide a power of 1.74. It can be shown that this exponent produces the best least squares fit to the trigonometric functions. Under these conditions the error resulting from the approximation will not exceed 1.35 per cent. The angles which may be resolved, however, will be subject to the restrictions indicated in the table.

An improved quarter-square multiplier circuit is shown in Fig. 7 (p. 97). Balance potentiometers have been added to the squaring circuits in order that the forward resistance of the diodes may be precisely matched, thereby preventing nonsymmetry in the squaring operation. This type of multiplier has several attractive advantages worthy of consideration: 1) high input impedance; 2) no requirement for special equipment; 3) wide-band response; and 4) zero output for zero input, a difficult characteristic to achieve with conventional multiplier schemes. It should be noted that all elements comprising the multiplier are either linear or passive. As a result accuracy is in no way dependent upon active nonlinear devices.

TABLE I (*Cont'd*)

CIRCUIT CONFIGURATION	TRANSFER FUNCTION
10. 	 $e_o \cong K(\cos e_i)$ $-\pi/2 \leq e_i \leq \pi/2$
11. 	 $e_o \cong K(\sin e_i)$ $0 \leq e_i \leq \pi$



TES
* 50 ~ pots to balance forward resistance of diodes.
Δ IM ~ input pot to balance outputs of thyrite squaring circuits.
□ 10M ~ pot to adjust K.
All fixed resistors ± 0.25 per cent.
Amplifier drift should not exceed 0.005 volt referred to the input.

Fig. 7—Thyrite multiplier circuit.

There is no intention to imply that the configurations given in the table are inflexible or all inclusive. They are intended as a guide to what may be done and other configurations are possible. These may be added to the table as dictated by future requirements.

Conclusion

Thyrite resistors, properly chosen and adjusted by the addition of the proper linear resistance, provide the means for generating a large class of nonlinear functions at small expense and with very little complication. The technique is valuable in releasing more complicated and expensive function generating equipment for the more difficult tasks for which they were intended, thereby increasing the total function generating capabilities of any analog computing installation.

A Novel Type of Isograph (Algebraic Equation Solver)*

P. VENKATA RAO†

***Summary*—Polynomial equations occur quite often in physical d mathematical problems and require an exact solution. Several blems in control engineering, aerodynamics, and many other ds require the formulation and exact solution of algebraic equans of quite a high order. In the past, several mechanical and elecal machines for the determination of the roots of polynomials have n developed, but all of them suffer from the serious disadvantage t they are very expensive and are incapable of high accuracy and idity of operation.**

This paper describes the development, design, and construction an inexpensive and portable isograph which is capable of locating roots of a polynomial with fairly high accuracy. This isograph can ve as a valuable aid to large-scale computing machinery, for which roots of a polynomial must be isolated before further calculations be performed. It can also serve as a very useful adjunct to other thematical processes such as complementary solution of differial equations, determination of latent roots of a matrix, and similar er applications.

It is hoped that this inexpensive and portable instrument will ve extremely useful in the design of servosystems and in other ds where the problem may be presented as a polynomial and that oretical studies of engineering problems will be greatly augnted by its use.

Introduction

T is well-known that polynomial equations occur quite often in physical and mathematical problems and require an exact solution. Several problems in servomechanisms, communication engineering, aerodynamics, and many other fields require the formulation and exact solution of algebraic equations of quite a high order. In the field of applied mechanics, all vibration problems involve the solution of a frequency equation which turns out to be a polynomial of degree corresponding to the degree of freedom enjoyed by the system. General analytical methods for obtaining the roots of equations as high as the fourth are available in standard textbooks on applied mathematics. However, Abel has proven that general solutions for polynomials of degree higher than the fourth cannot be obtained, and that particular solutions are the only ones possible in equations of the fifth degree and higher. Several numerical methods have been developed for the solution of higher degree algebraic equations, but each equation has to be treated as an individual case and its particular solution obtained. Some of the more well-known numerical methods are Ferrari's Method, Shih-Nge-Lin's Method, and Graffe's Method. Although these methods are quite useful for lower degree algebraic equations, they become extremely tedious and time consuming for equations of very high order, particularly if complex roots are involved. Consequently, several mechanical and electrical machines for the determination of the roots of polynomials have been developed, but all of them suffer from the disadvantage that they are very expensive and are incapable of high accuracy and rapid-

* Manuscript received by the PGEC, November 7, 1957; revised nuscript received, January 31, 1958.
† Dept. of Power Eng., Indian Inst. of Sci., Banalore, India.

ity of operation. Some of the more well-known machines are the Bell Telephone Laboratories Isograph, the Stibitz Isograph, the University of Texas Mechanical Synthesizer, Schooley's Isograph, and the Magslip Isograph of Parker and Williams. It is needless to point out that large and even medium present day digital computers are capable of solving very high-order algebraic equations in a very short time, with extraordinary accuracy.

Harmonic Synthesis

The design of most of the mechanical and electrical isographs, including the one described in this paper, is based on the principle of harmonic synthesis. The method of operation can best be explained by first transforming the general algebraic equation into the usual form in which it is generally set on the isograph.

A typical polynomial of high degree is:

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots a_nx^n = 0 \quad (1)$$

where the coefficients a_n are real numbers, whose value is less than unity. In general, the roots of the equation are of the form:

$$r = u + jv = re^{j\theta} = r(\cos\theta + j\sin\theta)$$

and there will be as many roots as the order of the equation.

By De Moivre's theorem $r = (u^2+v^2)^{1/2}$ and $\tan\theta = v/u$. Rewriting (1):

$$f(x) = a_0 + a_1r(\cos\theta + j\sin\theta) + a_2r^2(\cos 2\theta + j\sin 2\theta)$$
$$+ a_3r^3(\cos 3\theta + j\sin 3\theta) + \cdots a_nr^n(\cos n\theta + j\sin n\theta) = 0.$$

Separating, the real and imaginary parts, and equating each to zero

$$a_0 + a_1r\cos\theta + a_2r^2\cos 2\theta \cdots a_nr^n\cos n\theta = 0 \quad (2)$$

$$a_1r\sin\theta + a_2r^2\sin 2\theta + a_3r^3\sin 3\theta \cdots a_nr^n\sin n\theta = 0. \quad (3)$$

Multiply (2) by $\sin\omega t$ and (3) by $\cos\omega t$, where $\omega = 2\pi f$, f being an arbitrarily chosen independent variable. After adding these two products and collecting like terms in powers of r, one obtains

$$a_0\sin\omega t + a_1r\sin(\omega t + \theta) + a_2r^2\sin(\omega t + 2\theta)$$
$$+ a_3r^3\sin(\omega t + 3\theta) \cdots a_nr^n\sin(\omega t + n\theta) = 0. \quad (4)$$

Each of the terms of the above equation could be represented by a sine wave voltage, the modulus and phase angle of which can be varied over the desired range.

Basic Principle

The isographs developed by Schooley, Hart and Travis, and others consist of $n+1$, identical single-phase ac alternators of which n are provided with rotatable stators. All the alternators are driven by a motor on a common shaft in order to ensure that they operate at the same frequency. The rotatable stators are connected to a common shaft through gear ratios of 1:1, 2:1, 3:1, $\cdots$, n:1, so that as the hand crank is rotated slowly, the stator of the first alternator turns through θ, the second through 2θ, etc., down the line, to the nth stator which would turn through $n\theta$.

The output voltages of the various alternators would have the phase relations indicated in Fig. 1, and they are impressed across coefficient potentiometers set to the fractional values $a_1, a_2, a_3, \cdots, a_n$. The outputs of these coefficient potentiometers are then impressed across suitably ganged modulus potentiometers with tapped windings, in order to produce the various powers of the modulus. The correct phase relationship of the alternators, with reference to the first one, is thus adjusted by means of the knob on the hand crank whose calibration extends from zero to 360°.

In the actual operation of this type of isograph, an estimated value of the modulus is set on the ganged modulus potentiometers and the crank handle knob is slowly turned while observing the indicating instrument for a null or a minimum deflection. The outputs of the various ganged potentiometers are connected in series together with the voltage a_0 and the resulting voltage is read by the meter. By trial and error, a satisfactory null point can be obtained and the setting of the modulus potentiometer and the position of the knob of the hand crank would determine the modulus and argument of the root, respectively. This procedure should, of course, be repeated until all the roots of the polynomial have been determined. With this type of isograph it is also possible to solve equations with complex coefficients by setting the stator of each alternator to an initial phase displacement corresponding to the phase angle of the complex coefficient instead of at zero or 180° as is the case with ordinary positive and negative coefficients, respectively.

In the isograph developed and built by Marshall the different harmonic components are generated by using square generators of the rotating commutator type in conjunction with cascaded ganged potentiometers and unity gain isolating amplifiers. Filtering circuits shape the waves into sinusoidal form and potentiometers serve to bring in the coefficients a_n of the equation under study. The outputs are split electrically into the respective sine and cosine components. summed very accurately, and they are then displayed on a cathode-ray oscilloscope much as the mechanical isograph does it on a plain drawing board.

Choudhury's isograph, also designed on the principle of harmonic synthesis, is a purely electronic equipment without any mechanical moving parts. The different harmonic components are generated with the help of a delay line, fed from a matched generator, the frequency of which is swept by the time base voltage of the cathode ray oscillograph. The variation in frequency of the oscillator corresponds to the variation of the argument of the harmonic components. It is thus quite evident that, by controlling the amount of frequency sweep of the oscillator, any desired interval of the argument can be suitably expanded, thereby increasing the accuracy

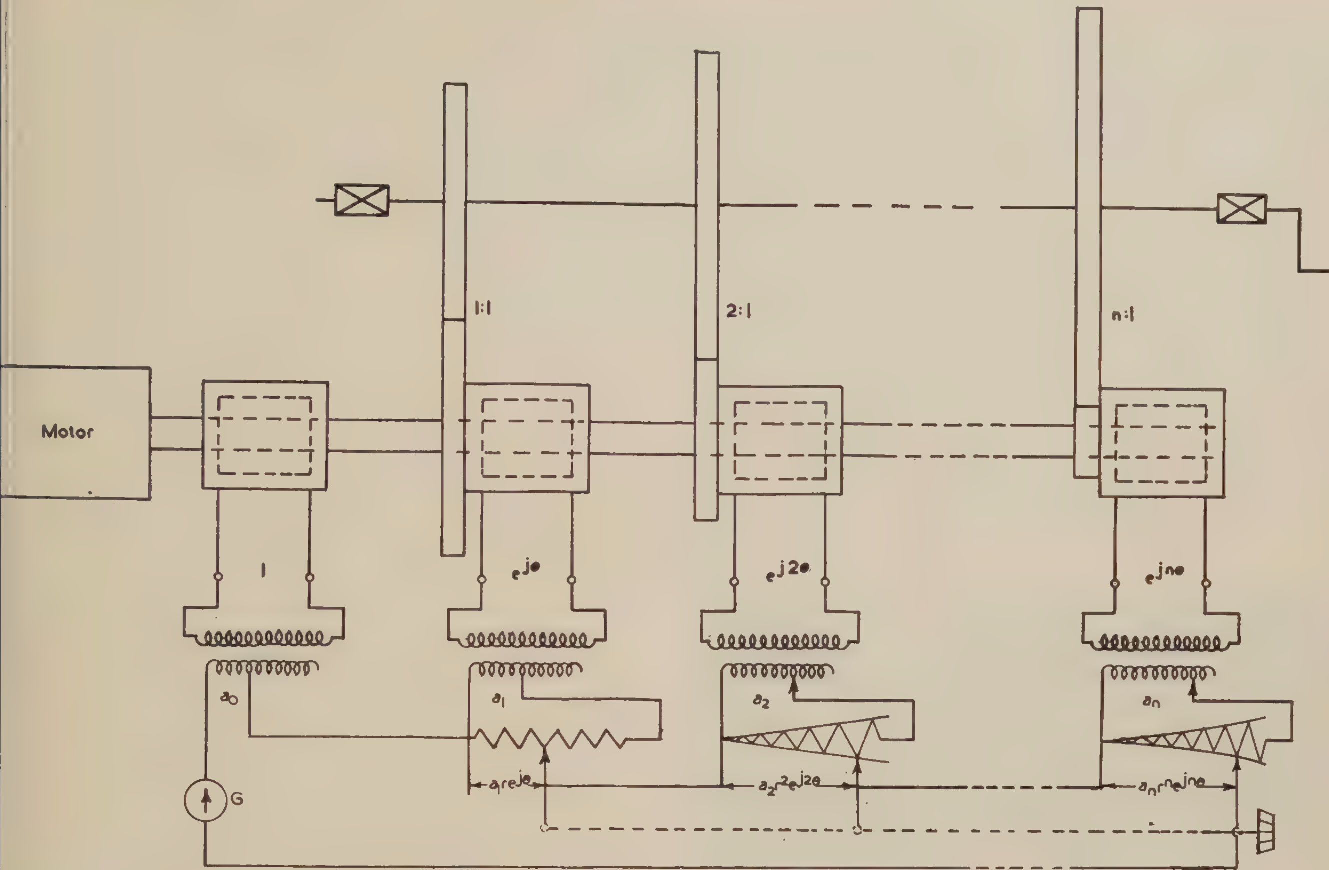


Fig. 1—Schematic arrangement of the isograph built by Schooley and others.

measurement. An interesting method of solving ;her degree algebraic equations was suggested re-ntly by Barker, who employed a transformer analog mputer of the Blackburn type. Bubb suggested the e of operational amplifiers for generating polynomials d determining their zeros.

In the Magslip isograph developed and constructed Parker and Williams the terms of the polynomials e simulated by ac voltages of varying phase obtained om Magslip transmitters excited by a balanced three-ase supply. The application of the principle of har-onic synthesis resulted in a fairly compact machine, pable of finding the roots with high accuracy. This ograph, like the earlier ones, could also be used for the lution of polynomials with complex coefficients and their evaluation at any point in the complex domain. ie Magslip isograph designed to solve sixth degree gebraic equations with real or complex coefficients cupies a standard rack 19 inches wide and 6 feet high, d the three-phase supply to the equipment is gen-ated by an oscillator the components of which are used in a similar rack of half the height.

Cumulative Phase Shifting

The design of the isograph described in this paper is sed on the principle of cumulative phase shifting for e generation of the various terms of the polynomials iose roots are to be determined. The application of this simple principle results in a compact, inexpensive, and portable instrument, the accuracy of which compares very favorably with that obtained by the very best of isographs built so far. As constructed, the isograph is capable of solving sixth degree equations having real coefficients, but the principle could be extended to deal with equations of higher degrees. It occupies a rack 19 inches wide and 16 inches high and it could be operated on the mains supply.

The block diagram of Fig. 2 illustrates the basic principle of working the isograph developed and built in the Indian Institute of Science. From the block diagram it is easily seen that the isograph essentially consists of a single type of unit which gives an output equal to r times the input but shifted in time phase by a controllable angle θ. The modulus of the root is set by ganged decade switches selecting the correct tap ratios of highly accurate three-winding transformers of special design, and the argument of the root is set by means of suitably ganged potentiometers in a set of simple RC phase shifter circuits. Referring now to (4) and Fig. 2, it can be seen that the isograph generates each one of the terms of the polynomial under study. Terms of the type $r^ne^{jn\theta}$ are obtained, simply by cascading the requisite number of units of identical design and providing control for r and θ with the aid of ganged switches. Of course, it is necessary to multiply these terms by the fractional values of the coefficients before the summa-

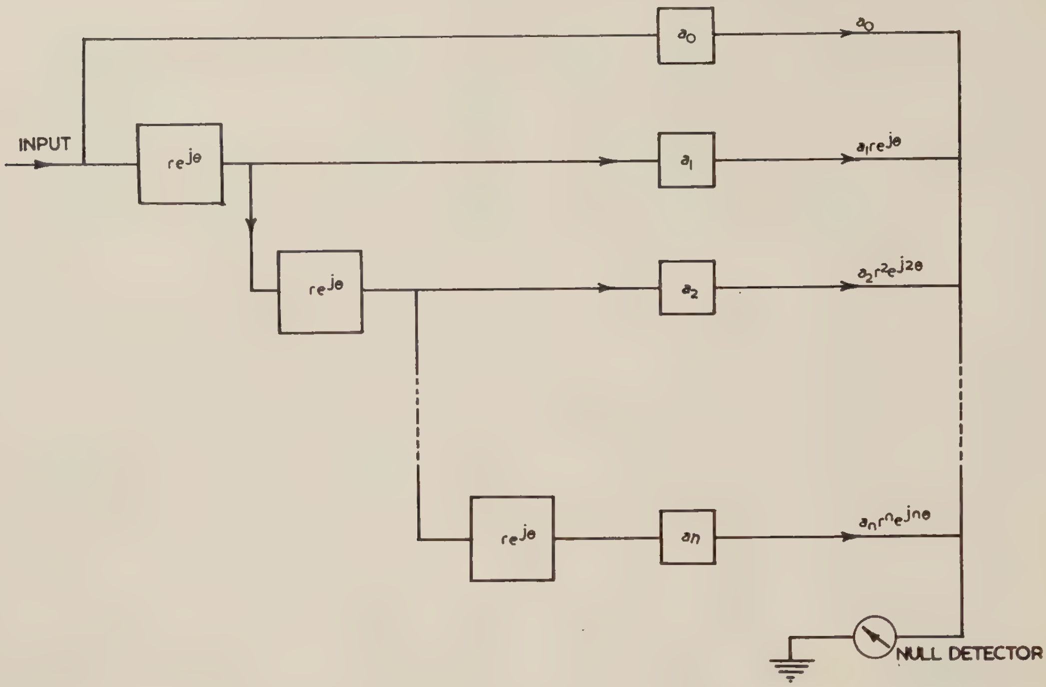


Fig. 2—Block diagram of the isograph described in this paper.

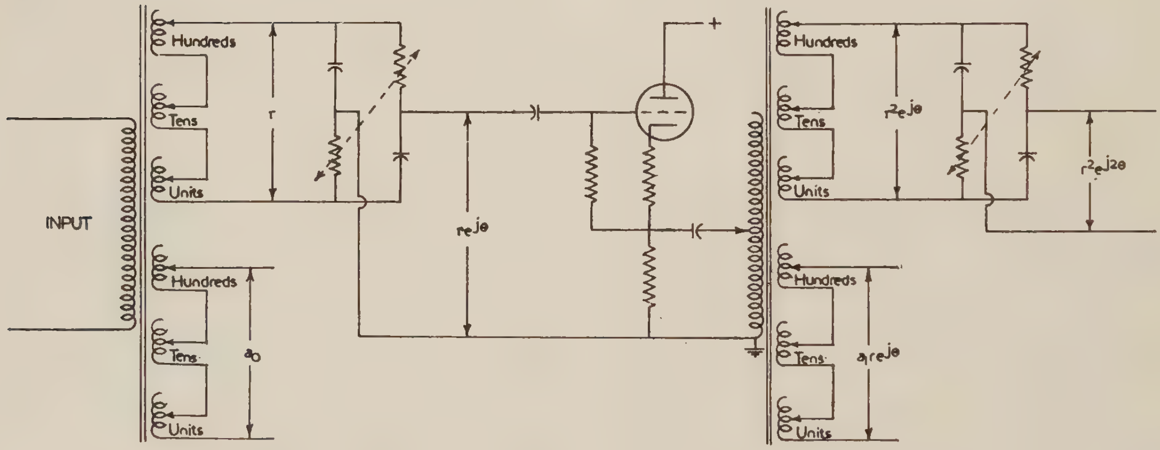


Fig. 3—Circuit arrangement of a typical complex multiplier unit.

tion of all the terms of the polynomial is effected by connecting the various voltages in series with a suitable measuring instrument. The mode of operation is thus to set the coefficients of the equation in the manner described below and to vary the modulus and phase angle of the variable until a zero reading is obtained. It is quite obvious that the isograph could also be used for the evaluation of a polynomial at any point in the complex domain.

The $re^{j\theta}$ Unit

Fig. 3 gives the actual circuit arrangement of a typical $re^{j\theta}$ unit. It essentially consists of a simple RC phase shifting network and a three-winding, multi-tapped transformer coupled by a cathode follower which acts merely as an impedance matching device. The equipment provides an output voltage which is exactly equal to r times the magnitude of the input voltage and at an angle θ degrees relative to the voltage applied to the input terminals. The modulus r is set as the turns ratio on the r winding of the transformer and θ is set as the indication of a calibrated potentiometer in a simple RC phase shifting network, consisting of two exactly identical capacitors in series with two ganged resistors connected in the manner shown, in Fig. 4(a). The vector diagram, in Fig. 4(b), shows the various voltages in the network; it can be seen that the output voltage is always equal to the input voltage but is displaced from the latter by an angle θ which can be changed from 0° when R is equal to zero to nearly 180° when R is equal to ∞.

The output voltage, therefore, can be set at any phase angle with reference to the input voltage merely by controlling the value of R with the aid of a ganged

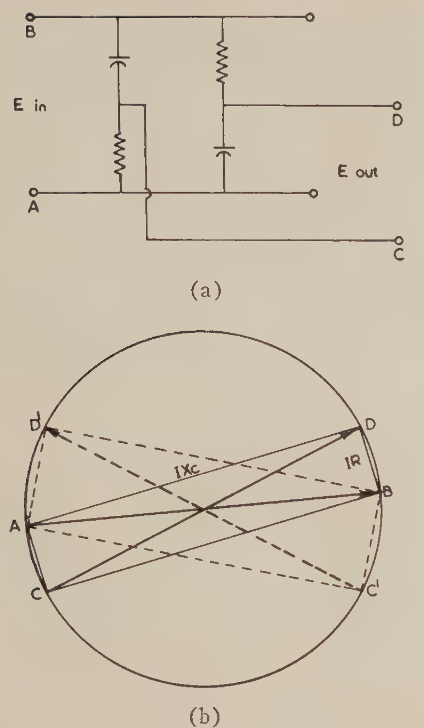


Fig. 4—(a) Circuit arrangement of the phase shifter.
(b) Vector diagram—phase shifter.

tentiometer. This voltage is then applied to the input rminals of a cathode-follower circuit which presents very high input impedance so that the performance of e phase shifter is not adversely affected. The output the cathode follower is impressed across the primary a three-winding transformer, built out of mumetal mpings with a fairly large area of cross section for e core. The primary winding is provided with a num- r of taps which can be adjusted over a small range d the two secondaries are tapped in decade arrange- nt so as to secure tap ratios from 0.001 to 0.999. The mber of turns of the primary winding is adjusted so at the voltage at a secondary of 2000 turns (which is o included for checking purposes) and the input tage to the phase shifter are exactly equal. These two tage signals are brought to a test panel at the back- e of the isograph in order to check frequently for the ity gain setting of the primary of the transformer. As well known the gain of a cathode follower is about 0.9 d minor variations of this, if any, can be compensated by varying the tap on the primary side of the output nsformer. This, evidently, does not affect the settings r and a. To maintain the accuracy of this unit over a le range of working voltages, regulation of the trans- mer is reduced to an absolute minimum by employing cially treated high permeability stampings and ting the base number of turns as 2000. As the maxi- m load on the output transformer corresponds to the pacitive reactance of the two capacitors in parallel hen R is set at zero), their value is chosen to be as high possible consistent with the availability of suitable ues of R to sweep θ through the range from 0 to 180°.

The above arrangement is, in effect, a highly stabilized amplifier of "unity gain" from the input of the phase shifter to the output at the 2000-turn secondary winding. Other values for the gain setting of the amplifier from 0.001 to 0.999 are secured independently from the two secondaries by means of decade switches. The appropriate fractional value of the coefficient is set on one of the secondaries. The setting on the other secondary corresponds to the modulus and it is manipulated by suitably ganged decade switches. The unit illustrated above is thus capable of producing two output voltages, $r^2e^{j2\theta}$ and $a_1re^{j\theta}$. The output of the r winding is applied to a similar arrangement consisting of a phase-shifter circuit and a three-winding transformer connected by a cathode follower. Fig. 5 is a photograph of the isograph designed to solve an algebraic equation of the sixth order, with real coefficients.

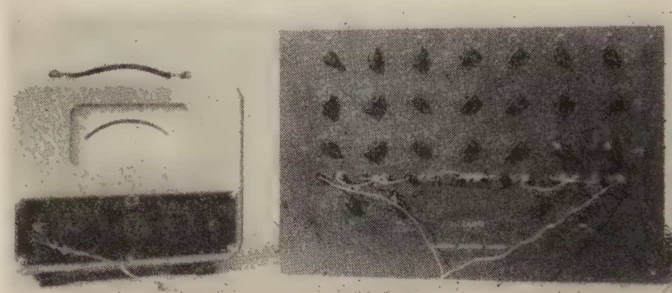


Fig. 5—Photograph of the isograph described in this paper.

Root Finding with the Isograph

As in the case of all the other isographs described in technical literature, practical design considerations have made it necessary to limit the range of the modulus to values less than unity. The range of variation of the argument, however, is limited to 0–180°, and by providing reversing switches on the primaries of the transformers it is possible to sweep θ through 0–360°. In order to bring all the coefficients within the compass of the isograph dials, it is usually necessary to divide all the terms of (1) by the largest coefficient or by some other convenient number, so that all the coefficients assume fractional values of comparable magnitude. The procedure for determining the roots whose modulus is less than unity is quite simple and straightforward. To take into account all those roots whose modulus is greater than unity, it is very necessary first to transpose the equation so that the reciproval of r appears in the transformed equation. We make the transformation

$$X = 1/x.$$

By making the simple substitution and multiplying all the terms of the polynomial by X^n, one gets

$$a_0 + \frac{a_1}{X} + \frac{a_2}{X^2} \cdots \frac{a_n}{X_n} 0; \qquad a_0X^n + a_1X^{n-1} \cdots a_n = 0.$$

This equation can be written down by inspection merely by inverting the order of the coefficients and introduc-

ing a new independent variable X which is the reciprocal of the actual root. This modified equation can now be examined for roots in the usual manner and the range $0<|x|<1$ for the modified equation corresponds to the range $1<|r|<\infty$ in the original equation.

In the case of all the polynomials where the ratio of the largest to the smallest coefficients is extraordinarily high, a simple substitution of the type $z=kx$ may be made with advantage.

$$f(kx) = f(z) = a_0 + \frac{a_1 z}{k} + \frac{a_2 z^2}{k^2} + \frac{c_3 z^3}{k^3} \cdots \frac{a^n z^n}{k^n} = 0.$$

The scaling factor k is selected in such a way as to reduce the very wide disparity in magnitude of the coefficients of the given equation. The number of complex roots will be equal to the difference between the order of the equation and the number of real roots. Furthermore, if the coefficients are real, as they are in the large majority of problems in servomechanisms and communication engineering, the complex roots must necessarily appear in conjugate pairs, so that only the upper half of the complex plane needs scanning. It is largely for this reason that provision is made in this isograph for a variation of 0–180° only, for the argument.

The actual method of operation of the isograph is thus to set up the coefficients of the equation on the panel and then to vary the modulus and argument in a judicious manner until a zero reading is obtained on the indicating instrument. In actual use, θ is set at zero and a search is made for positive real roots, r being varied until a null reading is obtained on the display meter. Negative real roots are obtained in an exactly similar fashion by setting θ at 180°. For the determination of complex roots, it is found in practice that it would be most convenient to vary one control, usually θ, until a minimum reading is obtained and then to vary the other control to reduce the minimum to zero if possible. In this way a root is approached fairly rapidly through successive approximations of the argument and modulus. The indicating instrument would give a sharply rising or falling indication with each change in either of the two controls signifying the approach to or retreat from a zone containing a root.

Measuring Instrument

When the settings of the modulus and argument switches correspond to a root, the fundamental components of the voltage balance out; however, their harmonic components give a residual voltage, as the phase shift, introduced for different harmonics, would be different. In order to be able to measure the signal strength to the high accuracy desired, it is absolutely necessary to filter out all traces of harmonic distortion from the output signal. This can be achieved by incorporating a passive discriminating filter which effectively attenuates the harmonics and passes the fundamental without any appreciable change in its magnitude. A more satisfactory discrimination, however, can be secured by a frequency selective amplifier which gives an output with a very high amplification for the desired frequency and little or no amplification for the harmonic components.

The output of this amplifier is fed back to the input, through a twin-T filter. This filter is of the rejector type and passes all the harmonics without much attenuation except the fundamental. The negative feedback is operative very largely for the harmonics, while it is ineffective for the fundamental. In the frequency selective amplifier built here, for use in conjunction with a tube voltmeter, the discrimination is of the order of 100:1 so that the relative amplification of the fundamental is about 100.

Complex Coefficients

An important extension that has interesting possibilities is the solution of algebraic equations involving complex coefficients. It is obviously necessary to connect an additional phase shifting network to the output side of the coefficient winding of the transformer of each of the $re^{i\theta}$ units. A very practical use for complex coefficients is in the transformation of the point of origin in the complex plane to some other convenient point, approximately located in the neighborhood of a cluster of complex roots. By treating this new point as the origin, in exploring for the various roots located as described, their determination is greatly simplified.

Conclusions

The isograph described in this paper is capable of locating the roots of a polynomial with high accuracy and in a short time of about one hour, provided that the roots are fairly well separated. In equations having closely spaced roots the minima obtained on the meter are rather broad and it is comparatively difficult to determine the roots to a high degree of accuracy, as it is even in the case of the isographs developed earlier. With the aid of a small amount of computational work on a desk calculating machine, it is possible to find all the roots of an algebraic equation correct to the fourth decimal place. With the isograph itself a precision of about 0.8 per cent can easily be achieved. The isograph is capable of yielding solutions with an accuracy sufficiently great for most engineering problems and with a rapidity of operation far exceeding that of any other device developed so far.

The isograph also lends itself to several other applications as well. It can serve as a valuable aid to large-scale computing machinery, for which the roots of a polynomial must be isolated before further calculations can be performed. It can also serve as a very useful adjunct to other mathematical processes, such as complementary solution of differential equations, determination of latent roots of a matrix, and similar applications.

It is hoped that this inexpensive and portable instrument will prove extremely useful in the design of servomechanisms and in other fields where the problem may be presented as a polynomial and that theoretical

studies of engineering problems will be greatly augmented by its use. It will also be of great value in the calculation of the transient behavior of linear electrical and mechanical systems.

Acknowledgment

The author is highly indebted to Professor C. S. Ghosh for his kind encouragement and constant help during the progress of this work. Help received from B. I. Gururaj, G. Krishna, and T. L. Venkatesh is acknowledged with sincere thanks. Thanks are also due to Professor A. Ramachandran for his kind interest in this work and to the authorities of the Indian Institute of Science for their permission to publish this paper.

Bibliography

[1] Abel, N. H. "Oeuvres Completis," *Christiania* (*Oslo*), Vol. 1 (1881), p. 28.
[2] Dietzold, R. L. "The Isograph, A Mechanical Root-Finder," *Bell Laboratories Record*, vol. 16 (1937–1938), pp. 130–134.
[3] Mercner, R. C. "The Mechanism of the Isograph," *Bell Laboratories Record*, Vol. 16 (1937–1938), pp. 135–140.
[4] Schooley, A. H. "An Electro-Mechanical Method of Solving Equations," *RCA Review*, Vol. 3 (1938), pp. 86–96.
[5] Hall, D. and Graham, S. "An Electrical Algebraic Equation Solver," *Review of Scientific Instruments*, Vol. 10 (1938), pp. 302–307.
[6] Hart, H. C., and Travis, I., "Mechanical Solution of Algebraic Equations," *Journal of Franklin Institute*, Vol. 225 (1938), pp. 63–72.
[7] Brown, S. L., and Wheeler, L. L. "A Mechanical Method for Graphical Solution of Polynomials," *Journal of Franklin Institute*, Vol. 231 (1941), pp. 223–243.
[8] Marshall, B. O. "The Electronic Isograph for Roots of Polynomials," *Journal of Applied Physics*, Vol. 21 (April, 1950), pp. 307–312.
[9] Bubb, F. W. "A Circuit for Generating Polynomials and Finding Their Zeros," PROCEEDINGS OF THE IRE, Vol. 39 (December, 1951), pp. 1556–1561.
[10] Fich, S. *Transient Analysis in Electrical Engineering*. New York: Prentice-Hall, Inc., 1951. See ch. 8.
[11] Soroka, W. W. *Analog Methods in Computation and Simulation*. New York: McGraw-Hill Book Co., Inc., 1954.
[12] Chowdhury, A. K. "The Electronic Root-Finder," *Indian Journal of Physics*, Vol. 29 (1955), pp. 468–473.
[13] Parker, G. M., and Williams, R. W. "A Magslip Isograph," *Journal of Scientific Instruments*, Vol. 32 (1955), pp. 332–338.
[14] Barker, J. R. "Application of a Transformer Analogue Computer," *British Journal of Applied Physics*, Vol. 7 (1956), pp. 303–308.
[15] Choudhury, A. K., and Nag, B. R. "R.C. Network Analogues," *Indian Journal of Physics*, Vol. 31 (1957), pp. 121–134.

Logically Micro-Programmed Computers*

JOHN V. BLANKENBAKER†

Summary—**Extremely simple digital computers exploiting the concepts of simulation and micro-programming are described. Logical rather than arithmetic micro-programming operations are employed for generality and greater simplicity. The resulting class of computers can duplicate the behavior of any finite state digital computer except in solution time. Also, design techniques are given for computers employing only multiple-bit time delays.**

Introduction

If one digital computer design requires less components by virtue of its structure or form than another functionally equivalent computer requires, the first computer is potentially preferable on the basis of cost and reliability. In designing for an application with a fixed problem, it would be desirable to keep the computer as simple as is consistent with the required solution speed. In many problems this is very low.

Unfortunately, it is not always clear how to achieve the simpler forms. In especially simple applications more computing capacity may be postulated than is necessary and it is then difficult to see how the structure may be reduced. Experience in the computer arts has shown that it is generally easier to add additional structure to a design to increase its computational ability than it is to accomplish the converse of simplifying the equipment with a reduction of its computing capacity.

One simple form of digital computer, which might be used as the basis for more complex designs, is described. In its simplest form the area of application would be small. With additional structure, however, the range of problems amenable to solution can be extended. While some brief suggestions are given for means of increasing the computational ability of the machine, these thoughts are not pursued in detail.

The Basic Computer

The first computer is described by (1). The logical notation employed is "+" for the nonexclusive OR, "·" for AND (sometimes omitted), and "—" for the complement of a variable or function.[1] Subscripts identify time intervals. Synchronous techniques are assumed in the serial computer, Fig. 1. L is a one-bit time delay, R is an n-bit time delay capable of storing n bits of information, and similarly M and N are each mn

* Manuscript received by the PGEC, December 16, 1957; revised manuscript received, April 2, 1958.
† Dept. of Elec. Eng., Mass. Inst. Tech., Cambridge, Mass.

[1] E. C. Nelson, "An algebraic theory for use in digital computer design," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-3, pp. 12–21, September, 1954.

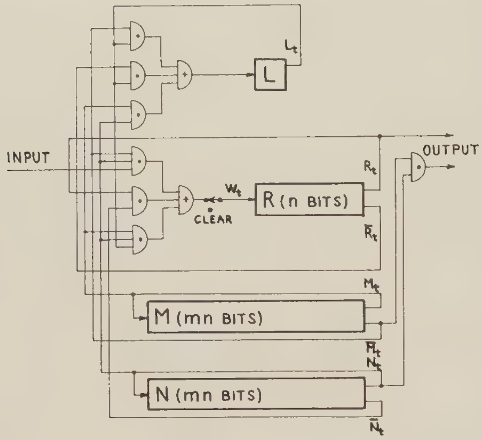


Fig. 1—Basic computer.

bit time delays. L_t, R_t, M_t, N_t are the "true" outputs of their respective devices, while W_t is the input to R. At any one time M_t and N_t describe one of the four micro-instructions which the machine can execute. The *M-N* sequences of *mn* "0"s and "1"s do not change for a given set of computer characteristics.

$$L_{t+1} = L_t \cdot \overline{M}_t + L_t \cdot \overline{R}_t + M_t \cdot N_t \tag{1a}$$

$$R_{t+n} = W_t = \overline{M}_t \cdot N_t \cdot \text{Input} + \overline{N}_t \cdot R_t + M_t N_t \cdot L_t \tag{1b}$$

$$M_t = M_{t-mn} \tag{1c}$$

$$N_t = N_{t-mn} \tag{1d}$$

$$\text{Output} = R_t, \quad \text{when} \quad \overline{M}_t \cdot N_t = 1. \tag{1e}$$

The four micro-instructions are given in Table I. As a result of the *do nothing* instruction, L retains its present state and the input to R is equal to the output bit of R.

TABLE I

INSTRUCTION CODE FOR BASIC COMPUTER

Name	Symbol	Code M	Code N	Function
Do nothing	—	0	0	$W_t = R_t$, $L_{t+1} = L_t$
Input-Output	Z	0	1	W_t = input, output $= R_t$, $L_{t+1} = L_t$
Logical product	O	1	0	$W_t = R_t$, $L_{t+1} = L_t \cdot \overline{R}_t$
Store	W	1	1	$W_t = L_t$, $L_{t+1} =$ "1"

The input bit from external sources is entered into R and one bit of the computer output is taken from R_t in the *input-output* instruction. The logical product of L_t and $\overline{R}_t$ is formed and entered into L in *logical product*. In *store* a bit equal to L_t is entered into R and L_{t+1} is determined as a "1".

PROGRAMMING

Programming usually consists of two steps. For a given problem one approach might be to program the LMC (Logically Micro-programmed Computer) as a general purpose "arithmetic" computer. In general in this first step, an *M-N* program is found which causes the LMC to simulate another digital machine. This *M-N* program of micro-instructions constitutes a description of a number of Boolean algebraic equations describing the simulated machine which the LMC evaluates. If the LMC is to have permanently the same *M-N* program, this may be entered into the machine when it is built; otherwise an *M-N* program loading mechanism is required.

As the second step the simulated machine is programmed, presumably by use of a more or less conventional instruction code. The state of the simulated computer as given by its memory elements is represented in the LMC by the information in R. Programming this simulated computer consists of finding the initial state of its memory bits. Since there are adequate references on this second part of programming, no discussion will be given concerning it.

Programming will be described by way of a few specialized examples. In the way of notation the following is introduced. Then bit positions of R, which circulate, are designated as $P1, P2, \cdots, Pn$. Time is divided into major and minor cycles corresponding respectively to the circulation times of the *M-N* channels and of R or the variable loop. Each bit time of a major cycle may be denoted by two numbers. From a time, which is taken to be coincident with the appearance of $P1$ at R_t, the bit times are numbered from 1 to n corresponding to the variable position number. This will be the second half of the time number. The first half of the time number denotes the minor cycle and runs from 1 to m. Thus, $t(6,4)$ denotes the fourth bit time of the sixth minor cycle. Any micro-instruction in the *M-N* program may be taken as the fiducial mark (if the P's are yet unnumbered).

The micro-instruction program can conveniently be written in the form of an n by m array. The n columns correspond to the bit times in a minor cycle (and to the variable position appearing at R_t), while rows of the array correspond to minor cycles. Entries are the micro-instructions.

The principal basic programming techniques are illustrated in the micro-instruction array of Table II. The micro-instruction occurring at $t(1,1)$ (see Table I for instruction symbols) has no part in the programming being illustrated. As a result, however, L is in the "1" state at $t(1,2)$. At this time the logical product instruction determines L_{t+1} as $L_t \cdot \overline{R}_t = 1 \cdot \overline{R}_t$; or the complement of the variable in $P2$ has been entered into L.

TABLE II*

PROGRAM OF MICRO-INSTRUCTION FOR LOGICAL COMPUTATION of $f(V2, V3, V5, V6) = \overline{V2} + V3 \cdot \overline{V5} \cdot \overline{V6}$

$m \backslash n$	1	2	3	4	5	6	7	8	9
1	W	O	—	—	—	—	—	—	—
2	—	W	O	W	—	—	—	—	—
3	—	—	—	O	O	O	W	—	—
4	—	O	—	—	—	—	O	W	—
5	—	—	—	—	—	—	—	O	W

* The nine n numbers are the R bit positions and also define the variables at time $t(1,1)$.

This value holds through a series of do nothing instructions until $t(2,2)$, when it is stored back into the $P2$ position of R. If originally the Pi position were said to store a variable named Vi, the variable now in the $P2$ position must be called $\overline{V2}$. A careful distinction must be made between the variable position and the variable therein.

The two bit time computation starting at $t(2,3)$ is similar to the previous one but differs in that $\overline{V3}$ is stored in the $P4$ position. This merely illustrates that two variables may be stored in R which are complementary. A slightly more complex computation of the AND function, $V3\cdot\overline{V5}\cdot\overline{V6}$, begins at $t(3,4)$. After the $t(3,4)$ instruction, L contains $V3$. The logical product instruction at $t(3,5)$ causes $L_t\cdot\overline{R}_t$, equal to $V3\cdot\overline{V5}$, to be entered into L. At $t(3,6)$ the logical product of $L_t=V3\cdot\overline{V5}$ and of $\overline{R}_t=\overline{V6}$ is formed and entered into L. Thus at $t(3,7)$ the value of the quantity stored into R is equal to $V3\cdot\overline{V5}\cdot\overline{V6}$. The ability to complement variables and evaluate AND functions has now been demonstrated.

The next two series illustrate the evaluation of OR functions from the previous two operations. The desired objective is to have a variable in $P9$ equal to $\overline{V2}+V3\cdot\overline{V5}\cdot\overline{V6}$. By Boolean manipulations

$$\overline{V2}+V3\cdot\overline{V5}\cdot\overline{V6}=\overline{(\overline{\overline{V2}})\cdot(\overline{V3\cdot\overline{V5}\cdot\overline{V6}})}, \qquad (2)$$

which removes all "+" signs by using AND and complementation operations. $P7$ at the present point in the computation contains $V3\cdot\overline{V5}\cdot\overline{V6}$. The series which begins in the program matrix at $t(3,8)$ and ends with $t(4,8)$ evaluates the right side of (2), except for the uppermost complementation sign. This final step is completed with the series ending at $t(5,9)$ so that the $P9$ variable is equal to the left-hand side of (2).

The elements of encoding, or the translation of Boolean algebraic equations into micro-instructions, have been illustrated by the previous examples. The set of equations which are to be encoded and which define the computer to be simulated may be obtained from an existing computer design[2] or they may be obtained by original work.[3,4]

Any digital computer can be cast in the form of a set of logical difference equations,

$$X_{s+1}^i=f^i(X_s^1, X_s^2, X_s^3, \cdots, X_s^p),\ i=1,2,\cdots,p \qquad (3)$$

where the X^i are one-bit delay elements as denoted by the subscripts $s+1$ and s [compare with (1a)]. This time difference (which can be of varying periods) is the reason for calling these equations difference equations. In distinction to the use of t for real time, s is used for simulated time. For the present purposes, the functions f^i represent combinations of the logical operators AND, OR, and complementation. Computer memory elements which do not seem to represent one-bit delays, for example R or the n-bit delay of the LMC computer, can be represented as a series of one-bit delays. External inputs can be treated as outputs of delays which have unknown inputs and the outputs can be represented as inputs to delays whose outputs are never used.

Once a set of equations of the form of (3) are obtained, they would be recast to eliminate the OR operation. This will not increase the number of difference equations. Taking (1a), merely by way of example, if the OR operation is eliminated, it would appear as

$$L_{s+1}=\overline{\overline{(\overline{L_s\cdot\overline{M}_s})\cdot(\overline{L_s\cdot\overline{R}_s})\cdot(\overline{M_s\cdot N_s})}}. \qquad (4)$$

This last equation is too complex to be evaluated directly. As implied in the example of Table II, additional variables are defined. In this case, they could be

$$A_s=L_s\cdot\overline{M}_s, B_s=L_s\cdot\overline{R}_s, C_s=M_s\cdot N_s \qquad (5a)$$

and also

$$D_s=\overline{A}_s\cdot\overline{B}_s\cdot\overline{C}_s. \qquad (5b)$$

Then,

$$L_{s+1}=\overline{D}_s. \qquad (6)$$

To preserve the time relationship of (4), an s subscript is used with the intermediate variables A, B, C, and D. A physical interpretation is that no delay is associated with formation of these variables. In microprogramming any single equation, no distinction is made between delay and nondelay equations. What is important is the order in which the equations are encoded and evaluated. Following the time when all variables may be said to have their s value, all nondelay variables must be computed before using the nondelay variables in computing a delay variable. Also, a series of nondelay equations must be evaluated in a sequence running from the s variables to the $s+1$ variables.

Delays could also have been assigned to (5a) and then (5b) and (6) would be nondelay equations for time $s+1$. Also, fractional delays could be assigned to each equation.

Another problem arises because all variables, in particular the delay variables, are not advanced simultaneously to their new or $s+1$ value. If the computation is at a point where all delay variables have their s value, these values cannot, in general, be destroyed or altered until all the $s+1$ values have been computed. Actually, this problem arises only if synchronous machines, which in effect have equal delays throughout, are being modeled. The design of asynchronous machines assumes that the delay times are arbitrary. Simulating an asynchronous machine on an LMC fixes the delays in ordered amounts corresponding to the order of computation. Thus, some restrictions which must be imposed

[2] S. P. Frankel, "The logical design of a simple general purpose computer," IRE Trans. on Electronic Computers, vol. EC-6, pp. 5–14; March, 1957.

[3] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Inst.*, vol. 257, p. 161, March, and p. 275, April, 1954.

[4] M. Phister, Jr., "Logical Design of Digital Computers," John Wiley and Sons, Inc., New York, N. Y.; 1958.

in general asynchronous design may be removed if the design is to be simulated on an LMC.

A simple solution exists for the problem of the unequal delays in the simulation of designs based on the premise of equal delays. Two sets of values, old and new, of the delay variables are stored whenever necessary. From the old values the new values are computed and stored. When all new values have been computed, these values are transcribed to the positions formerly occupied by the old values and become the old values for the next round of computation. This process would be repeated once per major cycle. It is more convenient to temporarily store the new values as complements. Thus, in (4) and (5b) one variable position would store L_s, the old value, and another position would hold $\overline{D}_s$, the complement of the new value.

The variables A, B, and C represent transient information which may be destroyed as soon as D, or in general the equivalent of the new value of a variable, has been formed. Thus, only a few variable positions are needed for all of the intermediate variables. A gross estimate on n would be thrice the number of equivalent one-bit delays in a simulated synchronous machine (old values, complements of old values, complements of new values) plus the maximum number of OR terms in any one equation (if the equations are written in normal disjunctive form). If an asynchronous machine is simulated, the number of variables would be roughly equal to twice the number of equivalent one-bit delays (old values, complements of old values) plus the maximum number of OR terms in any of the normal disjunctive equations. An estimate on m is harder to form since it depends on the particular program used. A gross estimate, probably high, sets m equal to n plus the number of repeated uses of intermediate variable positions. An allowance also must be made for the input-output routine.

Example 1

An example of a simple counter is programmed to illustrate the techniques. The physical problem is to count the number of cycles of a waveform and indicate every eighth one, Fig. 2.

A rather standard synchronous circuit for this purpose is given by (7). A, B, and C form the counter of eight. The count sequence is 000-001-010-011-100-101-110-111-000-etc., with 111 taken to be the eighth count. G and H form an input interlock to adapt the asynchronous input to the synchronous circuits. On $GH=10$ a count is made, on the other combinations of G and H no count is made.

$$G_{s+1} = (\text{Input})_s \tag{7a}$$

$$H_{s+1} = G_s \tag{7b}$$

$$A_{s+1} = (\overline{A}_s\cdot B_s\cdot C_s + A_s\overline{B}_s + A_s\cdot\overline{C}_s)\cdot G_s\cdot\overline{H}_s + (\overline{G}_s + H_s)A_s \tag{7c}$$

$$B_{s+1} = (B_s\cdot\overline{C}_s + \overline{B}_s C_s)\cdot G_s\cdot\overline{H}_s + (\overline{G}_s + H_s)\cdot B_s \tag{7d}$$

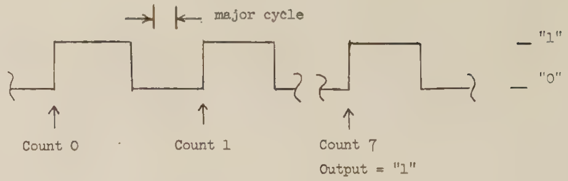


Fig. 2—Counter problem.

$$C_{s+1} = \overline{C}_s\cdot G_s\cdot\overline{H}_s + (\overline{G}_s + H_s)\cdot C_s \tag{7e}$$

$$(\text{Output})_{s+1} = A_s\cdot B_s\cdot C_s\cdot G_s\cdot\overline{H}_s. \tag{7f}$$

The variable register (R) in the LMC uses sixteen bit positions with the following variable assignment: $P1$, input (G_s) and output; $P2$, $\overline{G}_s$; $P3$, A_s; $P4$, $\overline{A}_s$; $P5$, B_s; $P6$, $\overline{B}_s$; $P7$, C_s; $P8$, $\overline{C}_s$; $P9$, $\overline{H}_s$; $P10$, $\overline{H}_s$; $P11$ through $P16$, working storage with $P13$, $P14$, $P15$, and $P16$ sometimes storing $\overline{H}_{s+1}$, $\overline{C}_{s+1}$, $\overline{B}_{s+1}$, and $\overline{A}_{s+1}$ respectively. A straightforward program matrix is given in Table III.

Variations

The basic computer can be modified in several ways. Instead of the logical product instruction, a logical sum could be used. The resulting equally simple instruction code would form the logical sum, $L_t+\overline{R}_t$. The store instruction would leave L_t in the "0" state. At some cost of convenience the Sheffer stroke function $(\overline{L}_t+\overline{R}_t)$ and the Pierce arrow function $(\overline{L}_t\cdot\overline{R}_t)$ could be used as the basic evaluating function. In these latter two cases, the store instruction would leave L in the "1" and "0" states, respectively.

A longer set of instructions can be included by using additional channels in parallel with the M-N channels. With a larger set of instructions additional delays, besides L, could be useful. Additional program channels and delays will not increase the flexibility of the computer, but would be introduced to increase solution speeds.

Another variation substitutes a multiple bit time delay for the one bit time delay L. In the basic computer L serves as a communication link between the binary variables in R. This function can also be performed by any multiple bit delay that is relatively prime to n. The case discussed here is for a delay of $n+1$ bit times. Denoting this delay of $n+1$ bits also by L, then (1a) for the basic computer would be replaced by

$$L_{t+n+1} = L_t\cdot\overline{M}_t + L_t\cdot\overline{R}_t + M_t\cdot N_t \tag{8}$$

and (1b) through (1e) would be essentially unchanged. An equivalent program for a given problem will differ in length so that a modification to (1c) and (1d) will be needed by way of changing the length of the M-N delays. The wired logic of the basic computer remains unchanged though and only the amounts of the various delays are changed. The instructions are the same and, in general, the same programming considerations apply.

A program written for the basic computer may be

TABLE III*

Program Matrix for Synchronous Eight Counter Counting Asynchronous Signals

Minor Cycle	G_s	$\overline{G}_s$	A_s	$\overline{A}_s$	B_s	$\overline{B}_s$	C_s	$\overline{C}_s$	H_s	$\overline{H}_s$	· · · working · · ·						Variable
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Var. position
1	Z	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	input/output
2	O	W	O	W	O	W	O	W	O	W	—	—	—	—	—	—	complements
3	—	O	O	—	—	O	—	O	O	—	W	—	—	—	—	—	$\overline{A}_sB_sC_sG_s\overline{H}_s$
4	—	O	—	O	O	—	—	—	O	—	—	W	—	—	—	—	$A_s\overline{B}_sG_s\overline{H}_s$
5	—	O	—	O	—	—	O	—	O	—	—	—	W	—	—	—	$A_s\overline{C}_sG_s\overline{H}_s$
6	O	—	—	O	—	—	—	—	—	—	—	—	—	W	—	—	$\overline{G}_sA_s$
7	—	—	—	O	—	—	—	—	—	O	—	—	—	—	W	—	H_sA_s
8	—	—	—	—	—	—	—	—	—	O	O	O	O	O	O	W	$\overline{A}_{s+1}$
9	—	O	—	—	—	O	O	—	O	—	W	—	—	—	—	—	$B_s\overline{C}_sG_s\overline{H}_s$
10	—	O	—	—	O	—	—	O	O	—	—	W	—	—	—	—	$\overline{B}_sC_sG_s\overline{H}_s$
11	O	—	—	—	—	O	—	—	—	—	—	—	W	—	—	—	$\overline{G}_sB_s$
12	—	—	—	—	—	O	—	—	—	O	—	—	—	W	—	—	H_sB_s
13	—	—	—	—	—	—	—	—	—	—	O	O	O	O	W	—	$\overline{B}_{s+1}$
14	—	O	—	—	—	—	O	—	O	—	W	—	—	—	—	—	$\overline{C}_sG_s\overline{H}_s$
15	O	—	—	—	—	—	—	O	—	—	—	W	—	—	—	—	$\overline{G}_sC_s$
16	—	—	—	—	—	—	—	O	—	O	—	—	W	—	—	—	H_sC_s
17	—	—	—	—	—	—	—	—	—	—	O	O	O	W	—	—	$\overline{C}_{s+1}$
18	O	—	—	—	—	—	—	—	—	—	—	—	W	—	—	—	$\overline{H}_{s+1}$
19	—	O	—	O	—	O	—	O	O	—	—	—	—	—	—	—	
20	W	—	—	—	—	—	—	—	—	—	—	—	O	—	—	—	output $(G_s\overline{H}_sA_sB_sC_s)$
21	—	—	—	—	—	—	—	—	W	—	—	—	—	O	—	—	$H_{s+1}\rightarrow H_s$
22	—	—	—	—	—	—	W	—	—	—	—	—	—	—	O	—	$C_{s+1}\rightarrow C_s$
23	—	—	—	—	W	—	—	—	—	—	—	—	—	—	—	O	$B_{s+1}\rightarrow B_s$
24	—	—	W	—	—	—	—	—	—	—	—	—	—	—	—	—	$A_{s+1}\rightarrow A_s$

* Some of the programmed equations, (7), are redundant. Minor cycles 1 and 24 may be combined.

lapted by the mechanical process of inserting n do othing instructions between each pair in the original rogram. This means though that only one bit in L ever of interest and the program is therefore very inficient. A more efficient procedure involves computing multaneously on up to $n+1$ evaluations. This limit sually cannot be reached due to timing difficulties; e., some variables cannot be computed until others ave been computed. A few more detailed comments are ven in the next example.

Example 2

The same problem as in the previous example is olved with the simulated counter taken to be asynhronous and with L taken to be an $n+1$ bit delay. ased on the coding of Table IV, then the following quations describe the counter.

$$G_{s+1} = (\text{Input})_s \tag{9a}$$

$$D_{s+1} = D_s\cdot\overline{G}_s + \overline{A}_s\cdot\overline{B}_s\cdot\overline{C}_s\cdot G_s + \overline{A}_s\cdot B_s\cdot C_s\cdot G_s + A_s\cdot B_s\cdot\overline{G}_s\cdot G_s + A_s\cdot\overline{B}_s\cdot C_s\cdot G_s \tag{9b}$$

$$C_{s+1} = C_s\cdot G_s + C_s\cdot\overline{D}_s + \overline{A}_s\cdot\overline{B}_s\cdot D_s\cdot\overline{G}_s + A_s\cdot B_s\cdot D_s\cdot\overline{G}_s \tag{9c}$$

$$B_{s+1} = B_s\cdot\overline{C}_s + B_s\cdot D_s + B_s\cdot G_s + \overline{A}_s\cdot C_s\cdot\overline{D}_s\cdot\overline{G}_s \tag{9d}$$

$$A_{s+1} = A_s\cdot C_s + A_s\cdot D_s + A_s\cdot G_s + B_s\cdot\overline{C}_s\cdot\overline{D}_s\cdot\overline{G}_s \tag{9e}$$

$$(\text{Output})_{s+1} = A_s\cdot\overline{B}_s\cdot\overline{C}_s\cdot D_s\cdot G_s. \tag{9f}$$

TABLE IV*

Asynchronous Coding of an Eight Counter

Present State				Next State, Input (G) 0				Next State, Input (G) 1			
A	B	C	D	A	B	C	D	A	B	C	D
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	1	0	0	0	1
0	0	1	1	0	0	1	1	0	0	1	0
0	0	1	0	0	1	1	0	0	0	1	0
0	1	1	0	0	1	1	0	0	1	1	1
0	1	1	1	0	1	0	1	0	1	1	1
0	1	0	1	0	1	0	1	0	1	0	0
0	1	0	0	1	1	0	0	0	1	0	0
1	1	0	0	1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	0	1	0	1	0	1	1	1	0
1	0	1	0	1	0	1	0	1	0	1	1
1	0	1	1	1	0	0	1	1	0	1	1
1	0	0	1	1	0	0	1	1	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0

* Output = "1" with $A_s\overline{B}_s\overline{C}_s\overline{D}_sG_s = 1$ or "1" with $A_{s+1}\overline{B}_s\overline{C}_s\overline{D}_sG_s = 1$.

For this counter the delay of R can be taken as 15 bits and that of L as 16 bits. The program matrix and variable assignments for these equations are given in Table V. It will be easier to interpret if the micro-instructions are read on a descending diagonal to the right. For example, at $t(5,2)$ the computation of $A_s\cdot G_s$ begins and resumes again at $t(6,3)$ with a logical product micro-instruction. This particular variable in L is po-

TABLE V

PROGRAM MATRIX FOR AN ASYNCHRONOUS EIGHT COUNTER

Minor Cycle	G_s	$\overline{G}_s$	$\overline{A}_s$	A_s	$\overline{B}_s$	B_s	$\overline{C}_s$	C_s	$\overline{D}_s$	D_s	· · · working · · ·					Variable
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Variable position
1	Z	—	—	—	—	—	—	—	—	—	—	—	—	—	—	input/output
2	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
3	—	W	—	—	—	—	—	—	—	—	—	—	—	—	—	$\overline{G}_s$
4	—	—	O	—	—	—	—	—	—	—	—	—	—	—	—	
5	O	O	O	—	—	—	—	—	—	—	—	—	—	—	—	
6	—	—	O	—	—	—	—	—	W	—	—	—	—	—	—	$\overline{D}_{s+1}$
7	—	—	—	—	—	—	—	—	O	—	—	—	—	—	—	
8	—	—	—	—	—	—	O	—	—	W	—	—	—	—	—	D_{s+1}
9	—	—	—	—	O	—	—	—	—	—	—	—	—	—	—	
10	O	O	—	—	—	—	—	—	—	—	—	—	—	—	—	
11	—	—	—	—	O	—	—	—	O	—	—	—	—	—	—	
12	—	—	—	—	O	—	—	O	—	—	W	—	—	—	—	A_sC_s
13	—	—	—	O	O	—	—	—	—	—	—	—	—	—	—	
14	O	O	—	—	—	—	—	O	—	O	—	W	—	—	—	A_sD_s
15	O	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
16	—	—	—	—	—	—	O	—	O	—	O	—	W	—	—	G_sA_s
17	O	—	O	O	—	—	—	—	—	—	W	O	—	—	—	$B_s\overline{C}_s$
18	—	—	—	—	—	—	O	—	—	—	—	—	O	W	—	$\overline{G}_sB_s\overline{C}_s\overline{D}_s$
19	—	O	—	—	O	O	O	—	—	O	—	W	—	O	—	B_sD_s
20	—	O	—	—	—	—	—	—	—	—	—	—	—	—	—	
21	—	O	—	O	—	—	—	—	—	O	O	—	W	—	—	G_sB_s
22	—	O	O	O	—	—	—	—	O	—	W	O	—	—	—	$C_s\overline{D}_s$
23	—	—	O	—	O	O	—	—	O	—	—	—	O	W	—	$\overline{G}_s\overline{A}_sC_s\overline{D}_s$
24	—	—	W	—	O	—	—	—	—	—	—	W	—	O	—	A_{s+1}, G_sC_s
25	—	O	O	—	—	—	O	O	O	—	—	—	—	—	—	
26	—	—	O	W	—	O	—	—	—	—	O	—	W	—	—	$\overline{G}_s\overline{A}_s\overline{B}_sD_s$
27	—	—	—	—	—	—	O	O	—	—	W	O	—	—	—	$\overline{G}_sD_s$
28	—	—	—	—	—	—	—	—	—	—	—	—	O	W	—	$\overline{G}_sA_sB_sD_s$
29	—	—	—	—	—	O	—	—	—	—	—	W	—	O	—	$\overline{A}_s\overline{B}_s\overline{C}_sG_s$
30	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
31	—	—	—	—	W	—	—	O	—	—	—	—	W	—	—	B_{s+1}, $\overline{A}_sB_sC_sG_s$
32	—	—	—	—	O	—	—	—	O	—	O	—	—	—	—	
33	—	—	—	—	—	W	—	—	—	—	—	O	—	W	—	B_{s+1}, $A_sB_s\overline{C}_sG_s$
34	—	—	—	—	—	—	—	—	—	—	—	—	O	—	—	
35	—	—	—	—	—	—	—	—	—	—	—	—	—	O	W	$A_s\overline{B}_sC_sG_s$
36	—	—	—	—	—	—	—	—	—	—	—	—	—	—	O	
37	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
38	—	—	—	—	—	—	W	—	—	—	—	—	—	—	—	$\overline{C}_{s+1}$
39	—	—	—	—	—	—	O	—	—	—	—	—	—	—	—	
40	W	—	—	—	—	—	—	W	—	—	—	—	—	—	—	$G_sA_s\overline{B}_s\overline{C}_sD_s$, C_{s+1}

tentially modifiable at $t(7,4)$, $t(8,5)$, $\cdots$, $t(15,12)$ but it is not. At $t(16,13)$, $A_s \cdot G_s$ is stored in $P13$ and the "corresponding position" in L is set to "1". During this process the evaluation of other variables is being made simultaneously. For example, during minor cycle 7 five variables in L are of interest. The micro-instruction located at $t(6,9)$ pertains to the previous major cycle of computation. It should be studied as the conclusion of the program.

DISCUSSION

The concept of micro-programming of Wilkes[5,6] and Mercer[7] enters by way of comparison. Mercer gives as a definition of micro-programming:

Micro-programming is the technique of designing the control circuits of an electronic digital computer to formally interpret and execute a given set of machine operations as an equivalent set of sequences of micro-operations, elementary operations that can be executed in one pulse time.

The computers of this article are micro-programmed in a broader sense than is given by this definition. Both the "arithmetic and control circuits" are micro-programmed using micro-operations. Most of the same conclusions as Mercer reaches may be made for the LMC. Micro-programmed machines do contain extreme versatility and flexibility for very little complexity in the circuits. The LMC are serial machines (not out of necessity, however) and a conclusion to the effect that they are inherently fast is not forthcoming. Wilkes and Mercer generally envision parallel machines.

The basic LMC is not a fast machine. Solution speed decreases as approximately the square of complexity of the problem. For comparison purposes the simulation of the SEAC characteristics is estimated to involve a speed differential of about 6×10^8. No conclusion as regards speed should be drawn about micro-programmed computers in general. The basic LMC can be much improved by more complex micro-instructions and by

[5] M. V. Wilkes and J. B. Stringer, "Micro-programming and the design of control circuits in an electronic digital computer," *Proc. Camb. Phil. Soc.*, pp. 230–238; April, 1953.

[6] M. V. Wilkes, "The best way to design an automatic calculating machine," *Proc. Manchester University Computer Inaugural Conf.*; July, 1951.

[7] R. J. Mercer, "Micro-programming," *J. Assoc. Comp. Mach.*, vol. 4, pp. 157–171; April, 1957.

ternate means of treating some classes of bits, such as e memory bits of the simulated machine.

The means of handling input and output is restrictive t general. Since only a single information transfer line used, all data must be time coded. Additional input d output equipment may be convenient.

The work of Smith with digital system simulation ould definitely be noted in which he uses the concept storing the state and logic of the simulated machine recorded information in the memory.[8] The particular sign of Smith is more complex and specialized in type simulation but the value of simulation to reduce comexity is recognized.

[8] W. E. Smith, "A digital system simulator," *Proc. Western Joint mp. Conf.*, pp. 31–36; February 26–28, 1957.

Perhaps one of the most significant results is the demonstration of how an arbitrary design may be implemented without single bit delays (*i.e.*, without flip-flops or toggles). Conceivably, at some frequencies gating and amplification functions may be mechanizable but single bit memory loops may be unreliable.

ACKNOWLEDGMENT

The author wishes to note that G. M. Amdahl and L. D. Amdahl have independently discovered very much the same computer. The author is particularly indebted to Mr. Amdahl for discussions about the computer. The work reported in this paper was done while the author was employed by The Ramo-Wooldridge Corporation, Los Angeles, Calif., during the summer of 1957.

Analytical Design of Resistor-Coupled Transistor Logical Circuits*

M. W. MARCOVITZ† AND E. SEIF†

Summary—The object of this paper is to analyze and to develop sign procedures for a resistor-coupled transistor circuit used in the echanization of logical operations. The basic circuit consists of e transistor and a number of resistors. This circuit performs the R function followed by the NOT function or the AND function llowed by the NOT function. With these compound functions echanized it is possible to build any logical system.

The first requirement for operation of this circuit is that the nsistor must be saturated if one or more inputs are low. The cond requirement is that the transistor must be cut off if all of the puts are high. A "worst case" analysis is performed for each of ese requirements.

Three types of solutions are described and discussed: general rpose, intermediate (flexible), and special purpose. The general rpose design uses one standard stage for the mechanization of an tire logical system. The special purpose design is tailored to fit a mplete logical system. If any logical change is made, a large part the circuitry must be redesigned. An intermediate (flexible) den uses different stage designs, having different numbers of inputs d outputs.

Procedures are developed for the design of the general purpose stem in two ways: hand computation with a slide rule; computation a small digital computer (Burroughs E101). Procedures are deloped for the design of the intermediate flexible system using hand mputation. Examples of the design procedures are shown and timized for minimum required transistor base current. Circuits ere constructed and tested to verify these design procedures.

* Manuscript received by the PGEC, December 17, 1957; revised nuscript received, March 24, 1958. This paper is an abbreviated m of a thesis submitted May, 1957, to the Faculty of the Moore hool of Elec. Eng., University of Pennsylvania, in partial fulfillent of the requirements for the degree of M.S. in electrical engi-

† Burroughs Corp. Res. Center, Paoli, Pa.

INTRODUCTION

THE purpose of this paper is to analyze and to develop design procedures for a transistor circuit which is capable of performing the three basic logical functions AND, OR, and NOT. This circuit has been described by Cole, Chien, and Propster[1] and by Rowe and Royer.[2–4] The circuit is shown in Fig. 1.

The base circuit must be so designed that the transistor is cut off if all m input voltages are zero or very nearly zero. If any one or more of the m input terminals is negative, then enough base current must be drawn to assure that the transistor operates in the region of "saturation." Its collector voltage will then be nearly zero.

The m inputs are connected to the collectors of previous stages and the n-output resistors are connected to the bases of succeeding stages. The stage under discussion will be cut off (OFF) if and only if all preceding stages are in saturation (ON) so that all m inputs are nearly zero. When the stage is OFF, its collector volt-

[1] C. T. Cole, Jr., K. L. Chien, and C. H. Propster, Jr., "A transistorized transcribing card punch," *Proc. Eastern Joint Computer Conf.*, pp. 80, 83; December 10–12, 1956.

[2] W. D. Rowe, "Transistor NOR circuit," Trans. Paper 57-195, presented at Winter General Meeting AIEE; January, 1957.

[3] W. D. Rowe and G. H. Royer, "Transistor NOR circuit design," Trans. Paper 57-196, presented at Winter General Meeting AIEE; January, 1957.

[4] W. D. Rowe, "The transistor NOR circuit," 1957 IRE WESCON CONVENTION RECORD, pt. 4, pp. 231–245; August 20–23, 1957.

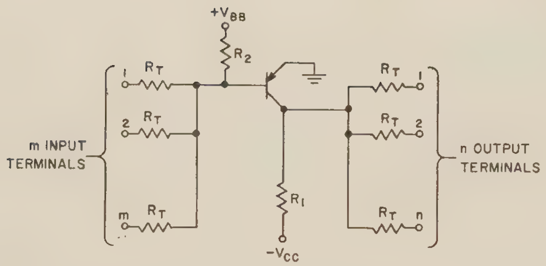


Fig. 1—The basic circuit.

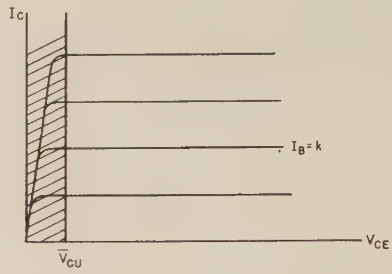


Fig. 2—Collector characteristics.

age will be sufficiently negative to assure that all n succeeding stages are ON. The stage under discussion will be ON if any one or more of the m-input stages are OFF. The collector of the stage under discussion will then be near zero volts and will not turn on any of the n succeeding stages. It should be noted, however, that a succeeding stage could, nevertheless, be ON if it has another input resistor connected to an OFF collector.

P-N-P transistors will be used for all examples in this paper and all definitions will be in terns of *p-n-p* transistors. In order to use *n-p-n* transistors, it is merely necessary to reverse the polarities of the power supplies.

Many of these stages can be interconnected to perform logical functions. The input resistors of any stage are the output resistors of preceding stages, and the output resistors of any stage are the input resistors of succeeding stages. Since every one of these resistors is both an input and an output resistor, it will be more convenient to refer to them as transfer resistors (R_T). (The definitions and symbols used in this paper are listed in Appendix I). The same circuit configuration is used to represent either an AND or an OR merely by redefining for each stage whether a conducting transistor shall represent the presence or the absence of a signal (*i.e.*, the signal or its inverse).

Design Criteria

A transistor can be in one of two states: OFF or ON, or in transition between them. By definition a transistor is said to be OFF if the emitter is reverse biased such that $I_C = I_{CO}$. I_{CO} does not change appreciably when the base-emitter reverse bias exceeds some minimum value, but is less than the breakdown voltage of the base-emitter junction. This minimum value is a characteristic of the transistor type to be used for a particular design and is available to the designer from the manufacturer's specifications or from test data. I_{CO} is similarly available. The first design criterion is that the base voltage of an OFF transistor must be equal to or more positive than some minimum required value obtained from transistor characteristics. The base voltage of a cutoff transistor is said to be "up" (*i.e.*, in the positive direction) and the symbol to be used is $+V_{BU}$ (*i.e.*, $V_{\text{Base, Up}}$). A minimum limit is indicated by a line below the symbol, a maximum limit by a line above the symbol. The minimum required value of base voltage for an OFF transistor is therefore $+\underline{V}_{BU}$. (V_{BU} merely represents the magnitude.) The first design criterion is therefore expressed by: $V_{BU} \geq \underline{V}_{BU}$, where $\underline{V}_{BU}$ is known from the characteristics.

When a transistor is ON it is required to be "saturated." For the purposes of this paper and by definition, a transistor is said to be saturated when its collector voltage is equal to or less negative than a value, $-V_{CU}$ ($V_{\text{Collector, Up}}$). The region of saturation is shown shaded in Fig. 2. The second design criterion is that the collector voltage of an ON transistor must be equal to or less negative than some maximum permissible value, $V_{CU} \leq \overline{V}_{CU}$.

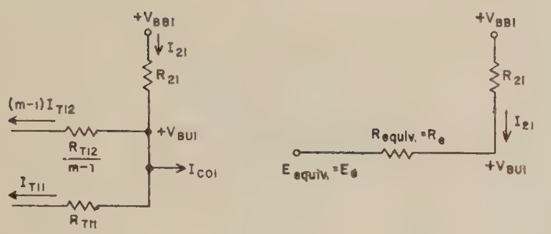
Other criteria may be imposed by the transistor specifications, such as maximum allowable collector voltage, emitter current, power dissipation, etc.

Worst Case Analysis

Every design must satisfy the two design criteria, $V_{BU} \geq \underline{V}_{BU}$ and $V_{CU} \leq \overline{V}_{CU}$, in order to assure satisfactory circuit operation. It is insufficient to know that the design center values of a particular design satisfy these conditions. All resistance values and voltage supplies may vary between maximum and minimum limits. A reliable design requires that each design criterion be satisfied even when all circuit parameters have simultaneously deviated by the maximum expected amount in the direction making it most difficult to satisfy the circuit criterion. A circuit in which all parameters have deviated in this manner is called a "worst case." Since there are two criteria there are two worst cases:

1) The worst case for V_{BU}. In this case every circuit parameter has deviated by the maximum expected amount in that direction which will tend to make V_{BU} smaller. If for this worst case V_{BU} is set equal to $\underline{V}_{BU}$, then V_{BU} will be larger than $\underline{V}_{BU}$ for all other circuit conditions. Therefore, $V_{BU} \geq \underline{V}_{BU}$.
2) The worst case for V_{CU}. In this case every circuit parameter has deviated by the maximum expected amount in that direction which will tend to make V_{CU} larger. If for this worst case V_{CU} is set equal to $\overline{V}_{CU}$, then V_{CU} will be smaller than $\overline{V}_{CU}$ for all other circuit conditions. Therefore, $V_{CU} \leq \overline{V}_{CU}$.

If a circuit satisfies both worst cases it is reliable and will never fail to meet the design criteria unless one or more parameters deviate *more* than their maximum



(a) (b)

Fig. 3—The V_{BU1} node.

xpected deviation. This is classified as a component ilure. Circuit diagrams describing the two "worst ase" conditions will be derived. For each worst case it ill be necessary to determine the direction in which ach circuit parameter must deviate in order to conribute to circuit failure.

Vorst Case for V_{BU}

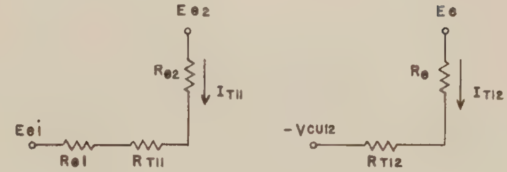
The results of this analysis are shown in Fig. 7 and eference may be made to that figure for the following aragraphs.

The output variable is V_{BU1} and it is necessary to xamine each independent variable to determine wheth- that variable should be a maximum or a minimum order to make V_{BU1} smaller. This can be done by exressing V_{BU1} as a function of all independent variables the circuit and obtaining the partial derivative of ach variable with respect to V_{BU1}. If this derivative is ositive for a particular variable, then that variable ust be a minimum to make V_{BU1} smaller; if the derivaive is negative, then that variable must be a maximum make V_{BU1} smaller. For a fairly elaborate circuit this rocedure can be very laborious. It is possible to examhe the circuit one branch at a time, starting at V_{BU1}, nd thereby greatly reduce the complexity of solution. This rocedure is followed here. The first variables to be exmined are V_{BB1} and R_{21}. For this purpose the rest of the ircuit, which is connected to V_{BU1} through I_{CO1}, I_{T11}, and $m-1)$ I_{T12}, Fig. 3(a), can be replaced by an equivalent ircuit, Fig. 3(b). E_e may, in general, be either positive r negative. The equivalent circuit is independent of R_{21} nd V_{BB1}. $E_e \neq fn(R_{21}, V_{BB1})$, $R_e \neq fn(R_{21}, V_{BB1})$. If I_{21} ows in the direction shown, $(V_{BB1}-E_e)>0$.

$$V_{BU1} = V_{BB1} - \frac{R_{21}}{R_{21}+R_e}(V_{BB1}-E_e). \tag{1}$$

$$\frac{\partial V_{BU1}}{\partial V_{BB1}} = 1 - \frac{R_{21}}{R_{21}+R_e} > 0. \tag{2}$$

herefore, V_{BU1} decreases as V_{BB1} decreases. Since for he worst case for V_{BU1} all variables deviate so as to ake V_{BU1} smaller, V_{BB1} should decrease and $\underline{V}_{BB1}$ is the orst case. (This is only the effect of V_{BB} acting on the ircuit through R_{21}. The effect of V_{BB} acting through R_{22} and R_{23} remains to be investigated.)



(a) (b)

Fig. 4—Equivalent circuits for I_{T11} and I_{T12} branches.

$$\frac{\partial V_{BU1}}{\partial R_{21}} = -(V_{BB1}-E_e)\frac{R_e}{(R_{21}+R_e)^2} < 0 \tag{3}$$

$\therefore \overline{R}_{21}$ is the worst case for $\underline{V}_{BU1}$.

Since the worst case deviations of R_{21} and V_{BB1} have been determined from an equivalent circuit which represents the complete circuit, $\overline{R}_{21}$ and $\underline{V}_{BB1}$ can be considered as fixed from here on.

Now:

$$\underline{V}_{BU1} = \underline{V}_{BB1} - \overline{R}_{21}I_{21} \tag{4}$$

$$\left.\frac{\partial V_{BU1}}{\partial I_{21}}\right|_{\underline{V}_{BB1},\,\overline{R}_{21}} = -\overline{R}_{21} < 0 \tag{5}$$

The worst case for $\underline{V}_{BU1}$ requires $\overline{I}_{21}$.

$$\overline{I}_{21} = \overline{I}_{CO1} + \overline{I}_{T11} + (\overline{m}-1)\overline{I}_{T12}. \tag{6}$$

$\overline{I}_{CO1} = \overline{I}_{CO}$, a known transistor characteristic; $\overline{m} \equiv m$.

The I_{T11} branch is not connected to a terminal point of the over-all circuit and therefore requires equivalent circuits at both ends of R_{T11}.

From Fig. 4(a):

$$\overline{I}_{T11} = \frac{E_{e2}-E_{e1}}{R_{e1}+R_{e2}+\underline{R}_{T11}}. \tag{7}$$

From Fig. 4(b):

$$\overline{I}_{T12} = \frac{E_e+\overline{V}_{CU12}}{\underline{R}_{T12}+R_e}. \tag{8}$$

$\overline{I}_{T11}$ implies $\underline{R}_{T11}$. $\overline{I}_{T12}$ implies $\underline{R}_{T12}$ and $\overline{V}_{CU12}$.

$\overline{V}_{CU12} = \overline{V}_{CU11} \equiv \overline{V}_{CU1}$; and $\overline{I}_{T11} = \overline{I}_{T12} \equiv \overline{I}_{T1}$.

At this point it may be convenient to generalize on the procedure. The circuit branch under investigation is shown unchanged, with the rest of the circuit represented by equivalent circuits. In each case a variable is to be maximized or minimized as a function of deviations, not of the equivalent circuit, but of parameters of the branch under discussion. The variable is either the output variable or an intermediate variable determined from previous considerations (such as $\overline{I}_{T12}$ above). Since the circuits are very simple, the required deviations are easily determined mathematically. For example, if the current through the circuit is to be maximized (as in the I_{T12} circuit above), then the series resistance must be minimized and the voltage difference between

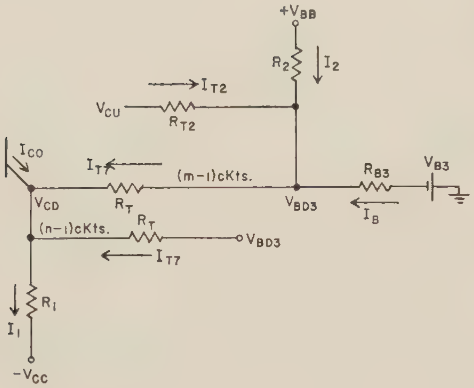


Fig. 5—Simplified $\overline{V}_{BD3}$ circuit.

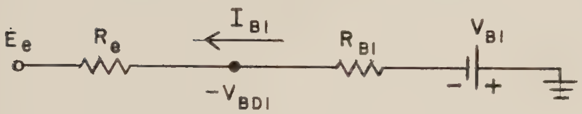


Fig. 6—The V_{BD1} node.

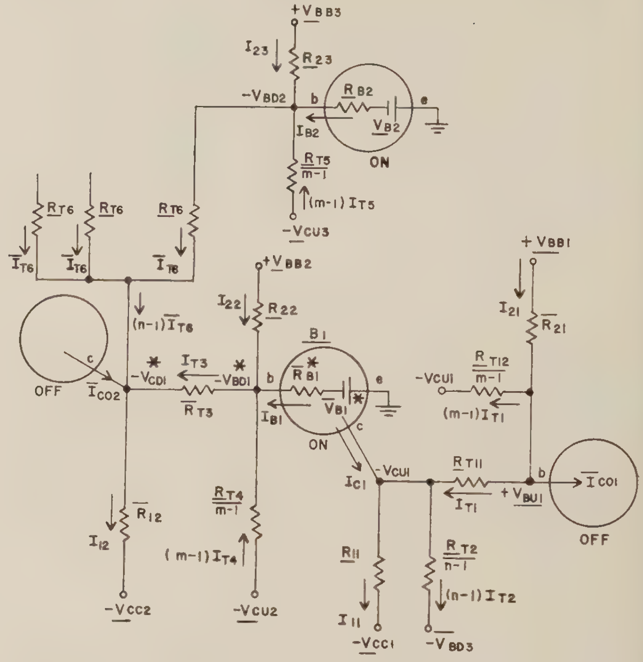


Fig. 7—The worst case for V_{BU}.

terminals must be maximized. Hence $\overline{I}_{T12}$ implies $\underline{R}_{T12}$ and $\overline{V}_{CU12}$.

Since the worst case deviations of the circuit up to this point have already been determined, the rest of the circuit can influence the output variable only through its influence on $\overline{I}_{T1}$.

$$\overline{I}_{T1} = \overline{I}_{11} + (\bar{n} - 1)\overline{I}_{T2} - \underline{I}_{C1}. \tag{9}$$

$\overline{I}_{11}$ implies $\underline{R}_{11}$ and $\overline{V}_{CC1}$, where V_{CC1} is the collector supply voltage as it influences the circuit through the R_{11} branch. V_{CC} also acts on the circuit through other branches and its influence on the over-all circuit therefore has not yet been determined.

$\overline{I}_{T2}$ implies $\underline{R}_{T2}$ and $\overline{V}_{BD3}$, where $\overline{V}_{BD3}$ is the maximum voltage at a conducting base which has one input resistor connected to a conducting collector and $(m-1)$ input resistors connected to cutoff collectors. A simplified circuit for the evaluation of $\overline{V}_{BD3}$ is shown in Fig. 5.

$$\underline{I}_{C1} = \underline{B}_1\underline{I}_{B1}. \tag{10}$$

$\underline{B}_1=\underline{B}$, a transistor characteristic.

To evaluate the worst conditions in the I_{B1} branch, an equivalent circuit is used to represent all of the rest of the circuit as viewed from V_{BD1} (Fig. 6).

For $\underline{I}_{B1}$ the circuit requires $\overline{R}_{B1}$ and $\overline{V}_{B1}$.

$$\underline{I}_{B1} = \underline{I}_{T3} - (\overline{m} - 1)\overline{I}_{T4} - \overline{I}_{22}. \tag{11}$$

$\overline{I}_{T4}$ implies $\underline{V}_{CU2}$ and $\underline{R}_{T4}$; $\overline{I}_{22}$ implies $\overline{V}_{BB2}$ and $\underline{R}_{22}$; $\underline{I}_{T3}$ implies $\overline{R}_{T3}$,and with $\overline{R}_{T3}$ determined:

$$\underline{I}_{T3} = \underline{I}_{12} - \overline{I}_{CO2} - (\bar{n} - 1)\overline{I}_{T6}. \tag{12}$$

$\overline{I}_{CO2}=\overline{I}_{CO}$, a transistor characteristic. $\underline{I}_{12}$ implies $\underline{V}_{CC2}$ and $\overline{R}_{12}$. $\overline{I}_{T6}$ implies $\underline{R}_{T6}$.

$$\overline{I}_{T6} = \overline{I}_{B2} + \overline{I}_{23} + (\overline{m} - 1)\overline{I}_{T5}. \tag{13}$$

$\overline{I}_{B2}$ implies $\underline{R}_{B2}$ and $\overline{V}_{B2}$. $\overline{I}_{23}$ implies $\overline{V}_{BB3}$ and $\underline{R}_{23}$. $\overline{I}_{T5}$ implies $\underline{V}_{CU3}$ and $\underline{R}_{T5}$.

In this analysis the worst case conditions for V_{BB} were found to be $\underline{V}_{BB1}$, $\overline{V}_{BB2}$ and $\overline{V}_{BB3}$; for V_{CC} they were found to be $\overline{V}_{CC1}$ and $\underline{V}_{CC2}$. The voltage supplies are, in general, common to the whole circuit. If $\overline{V}_{CC}$ and $\underline{V}_{CC}$ were used simultaneously, the resultant circuit requirements would be "worse than the worst case," which is inefficient, especially since the amount of extra safety would be unknown. More reliable circuits are designed if the worst case limits of the voltage supplies are analytically derived. For the worst case for V_{BU}, the limit values are $\underline{V}_{BB}$ and $\underline{V}_{CC}$. The worst case circuit for V_{BU} is shown in Fig. 7. The worst case circuit for V_{CU} can be derived in a similar manner and is shown in Fig. 8. The two circuits shown in Figs. 7 and 8 differ in only two circuit parameters: R_{T11} and V_{BB}.

Basic Equations

The worst case circuits (Figs. 7 and 8) yield the two basic equations for a general purpose system.

The basic equation for the worst case for V_{BU} is written about the $\underline{V}_{BU1}$ node of Fig. 7.

$$\overline{I}_{CO1} = [(1 - \sigma)V_{BB} - \underline{V}_{BU1}](1 - \rho)G_2 - (V_{CU1} + \underline{V}_{BU1})m(1 + \rho)G_T \tag{14}$$

where

$$G = \frac{1}{R}.$$

V_{CU1} is an intermediate variable. Parametric equations can be written to define V_{CU1} by completing the circuit solution.

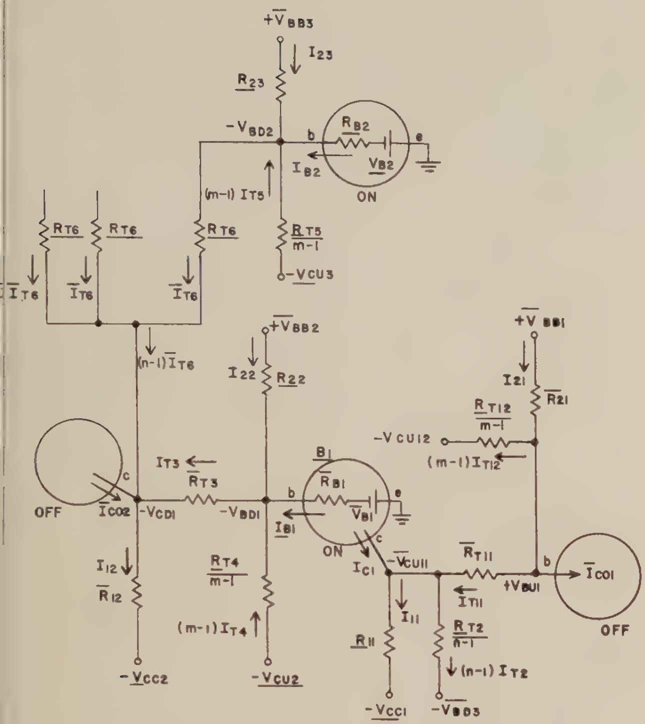


Fig. 8—The worst case for V_{CU}.

The basic equation for the worst case for V_{CU} is ritten about the $\overline{V}_{CU11}$ node of Fig. 8. It can be shown at $\overline{V}_{CU11} \doteq V_{CU12}$ and will hereafter be referred to as $_{CU1}$.

$$_1 = [(1-\sigma)V_{CC} - \overline{V}_{CU1}](1+\rho)G_1 + [(\overline{V}_{BD3} - \overline{V}_{CU1})(1+\rho)(n-1) - (V_{BU1} + \overline{V}_{CU1})(1-\rho)]G_T. \quad (15)$$

The parametric equations required for the solution the above equations were derived from the two worst se circuits. The following approximations were made: $_{BD1*} \doteq V_{BD1}$; $V_{CD1*} \doteq V_{CD1}$. V_{CD1} and V_{CD1*} are the oltages at the collectors of a cutoff transistor under orst case conditions. See Fig. 7 and Fig. 8.

From Fig. 8:

$$V_{BU1} = \frac{(1+\sigma)V_{BB}(1-\rho)G_2 - \overline{V}_{CU1}[(1-\rho) + (m-1)(1+\rho)]G_T - \overline{I}_{CO1}}{(1-\rho)G_2 + [(1-\rho) + (m-1)(1+\rho)]G_T}. \quad (16)$$

$$V_{CU1} = \overline{V}_{CU1}\frac{I_{B1}}{\underline{I}_{B1}} \text{ (see Appendix II).} \quad (17)$$

From Fig. 8:

$$_{B1} = [(V_{CD1} - V_{BD1})(1-\rho) - V_{BD1}(1+\rho)(m-1)]G_T - [V_{BD1} + (1+\sigma)V_{BB}](1+\rho)G_2. \quad (18)$$

From Fig. 7:

$$_{B1} = [(V_{CD1} - V_{BD1})(1-\rho) - V_{BD1}(1+\rho)(m-1)]G_T - [V_{BD1} + (1-\sigma)V_{BB}](1+\rho)G_2. \quad (19)$$

From Fig. 8:

$$\underline{I}_{C1} = \underline{B}I_{B1} \quad (20)$$

$$\overline{V}_{BD3} = \overline{V}_{B3} + \frac{\overline{R}_{B3}G_1G_T(m-1)(1-\sigma)V_{CC}}{(1-\rho)G_1 + n(1-\rho)G_T}. \quad (21)$$

From Fig. 8:

$$V_{BD1} = \overline{V}_{B1} + \underline{I}_{B1}\overline{R}_{B1}. \quad (22)$$

From Fig. 7:

$$V_{CD1} = \frac{V_{BD1}(1-\rho)G_T + (1-\sigma)V_{CC}(1-\rho)G_1 - \overline{I}_{CO2}}{[(n-1)(1+\rho) + (1-\rho)]G_T + (1-\rho)G_1}. \quad (23)$$

Substitution of the parametric equations in the two basic equations would result in complicated and nonlinear expressions. It is easier to handle the basic equations and parametric equations separately.

Types of Systems

There are three general types of systems: the general purpose system, the special purpose system and the flexible, intermediate system. The choice of system is determined by the nature of the application.

The General Purpose System

In this system a *standard* transistor stage is used. In some stages not all of the inputs and outputs are used; the unused transfer resistors are grounded. It is not necessarily required to use a general purpose stage with the maximum number of inputs and the maximum number of outputs to be found anywhere in the system. This may turn out to be unnecessarily wasteful of components, power, and space. It is often advantageous to choose a general purpose stage which will take care of the great majority of circuits and to use two stages in parallel (or emitter followers, or higher current gain transistors) for the few remaining circuits.

The Special Purpose System

In this system all of the logical circuitry must be designed by the *logical* designer before the *electrical* circuit design can begin. The output requirements must also be known. Each transistor stage is designed for optimum performance in the particular logical configuration in which it is employed. If any change is made in the logic after the circuit design has been completed, a large amount of circuit redesign might be required. This system is not covered in this paper because of its special purpose nature.

The Intermediate, Flexible System

An intermediate, flexible system has been developed which incorporates some of the advantages of both the special purpose and general purpose systems. In any system, logical elements will generally have different numbers of inputs and outputs. Circuit designs are developed for the various logical elements required. Assembly of these designed circuits into a system may require some modification as a function of the logical configuration. These modifications require a small amount of simple computation. An intermediate, flexible system can easily be changed to comply with required logical changes.

Design Procedures

Two basic equations (14) and (15) have been developed for the general purpose circuit to satisfy the two basic criteria. The 15 independent variables in these basic equations and the associated parametric equations, (16) through (23), are: two power supply voltages (V_{CC}, V_{BB}); power supply tolerance (σ); three resistors (R_1, R_2, R_T); resistor tolerance (ρ); six transistor parameters ($\underline{B}$, $\overline{I}_{CO}$, $\overline{R}_B$, $\overline{V}_B$, $\overline{V}_{CU}$, $\underline{V}_{BU}$); number of inputs (m); number of outputs (n). Solution of the two basic equations will determine only two unknown variables. Consequently, some other means of determining 13 variables must be established if the solution is to take this form.

Many of the independent variables may be assigned values based on practical engineering judgment and on the requirements of other parts of the device for which the logical circuitry is intended. The selection of a transistor type gives values for six of the variables. Selection of power supply voltages and tolerance, and resistor tolerance gives four variables. A knowledge of the logic of the circuit gives values for m and n. If all these values are assigned, the three resistors remain as the only three unknown independent variables.

General Purpose System

The following are some design procedures for a general purpose system.

Assume Values for Circuit Parameters: One design procedure is to assume values for all the circuit parameters except two. The two basic equations would then be solved for the two unknown parameters. If there is no solution new values would be assumed and a solution tried again. By assigning the independent variables as suggested above and by assigning a value to one of the resistors, the problem is reduced to the solution of the two basic equations with two unknowns. If there is no solution with the value of resistance assumed, a new value of resistance must be chosen and the solution repeated. There may be an infinite number of solutions, or there may be no solution, depending upon what values were assumed for the other circuit parameters; *e.g.*, m and n too high.

Assume Values for Intermediate Variables: The assignment of values to intermediate variables (*e.g.*, $\underline{I}_{B1}$, V_{CD1}) gives additional basic equations. These values would be selected on the basis of engineering judgment.

Optimum Solution: Since the general purpose solution involves the solving of two basic equations for the determination of more than two independent variables, there may be an infinite number of solutions. By establishing additional criteria it is possible to determine a more desirable or optimum solution. Some of the criteria that might be used for optimizing are: minimum transistor base current; minimum transistor power dissipation; minimum required transistor current gain ($\underline{B}$); minimum circuit power dissipation; optimum circuit for switching time considerations; maximum number of inputs (m); maximum number of outputs (n); maximum resistor tolerance (ρ); maximum power supply voltage tolerance (σ). Illustrations in this paper use minimum base current as an optimizing criterion since it will lead to low-transistor power dissipation and low-power supply requirements.

Special Purpose System

The above design procedures can also be used for the design of a special purpose system. Each logical element requires its own design. Since the load on the stage is known, a third equation is supplied.

Intermediate, Flexible System

The above design procedures for a general purpose system can also be used in designing an intermediate system. The two basic criteria must be satisfied. The load on a stage is known, giving a third equation. V_{CD1} should be a constant value for all circuits. Consequently, the assignment of a value for V_{CD1} gives a fourth basic equation.

Computer Solution of a General Purpose System

A solution of a resistor-coupled transistor logical element has been performed on a Burroughs E101 digital computer. An optimum solution was obtained with minimum transistor base current as a criterion.

The two basic criteria give the two basic equations (14) and (15). To optimize for minimum transistor base current ($\underline{I}_{B1}$), values are systematically assigned to $\underline{I}_{B1}$. Consequently, the equation for $\underline{I}_{B1}$ (18) is used as a third basic equation for this solution. A partial solution is performed to determine a minimum value for $\underline{I}_{B1}$ for which the two basic criteria are satisfied. The three conductances G_1, G_2, and G_T are solved as unknowns. All other independent variables are assigned values.

In order to simplify the solution, values are initially assigned to the intermediate variables $\overline{V}_{BD3}$, V_{CU1}, and V_{BU1}, since they have second order effects on the circuit solution. This eliminates the need for using parametric equations (16), (17), (19), and (21) in the solution of

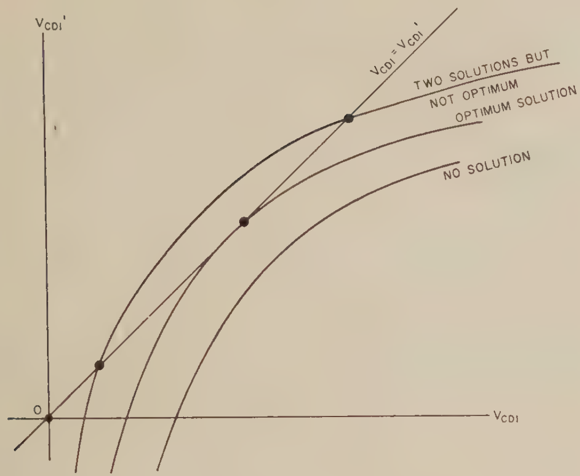


Fig. 9—Behavior of computer solution.

14), (15), and (18). Safe values are assumed as follows: $\overline{V}_{BD3} \doteq 2\overline{V}_{B3}$; $V_{CU1} \doteq \overline{V}_{CU1}$; $V_{BU1} \doteq \underline{V}_{BU1}$. The equations or $\underline{I}_{C1}$ (20) and V_{BD1} (22) are simple to calculate since $_{B1}$ is an assigned value. However, the equation for V_{CD1} (23) is a function of G_1 and G_T. An iterative rouine is used to solve (14), (15), and (18) with the paranetric equation for V_{CD1} (23). This involves an initial ssumption of a value for V_{CD1} and the direct solution f (14), (15), and (18) for G_1, G_2, and G_T. $V_{CD1'}$ is then alculated using (23). The computer selects a new value or an assumed V_{CD1} depending on the calculated $V_{CD1'}$ hich resulted from the previous trial value of V_{CD1}. alues of V_{CD1} are assumed such that the calculated alue of $V_{CD1'}$ approaches V_{CD1}. This process is coninued until it is determined if a solution exists for the ssigned value of $\underline{I}_{B1}$. The exact solution is defined by he intersection of a hyperbola and the line $V_{CD1} = V_{CD1'}$ Fig. 9). If a solution does exist, a lower value for $\underline{I}_{B1}$ is ssigned and a new solution performed. This procedure s continued until a minimum value for $\underline{I}_{B1}$ is deternined. The exact optimum value is defined as the point f tangency of the hyperbola with the line $V_{CD1} = V_{CD1'}$.

Using the calculated minimum $\underline{I}_{B1}$, values are calulated for V_{BU1}, V_{CU1}, and $\overline{V}_{BD3}$ with equations (16), 17), (19), and (21). With these values a more accurate ninimum $\underline{I}_{B1}$ is determined by using the computer soluion described above.

When the more accurate value of minimum $\underline{I}_{B1}$ is etermined, more accurate values of V_{BU1}, V_{CU1}, and $\overline{V}_{BD3}$ are calculated. With these values the solution is epeated to determine exact values for the three conluctances G_1, G_2, and G_T.

A number of circuits were designed using this comuter solution and the effects on the circuit design of he variation of circuit parameters were studied.

Hand Solution of a General Purpose System

This section describes the design of standard tranistor stages suitable for use in a general purpose sysem. The procedure will be to select a transistor type,

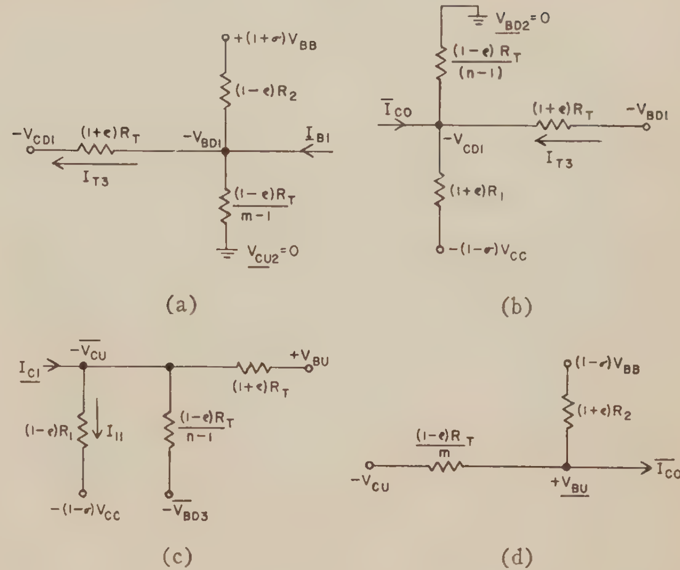


Fig. 10—Worst case circuits for manual solution.

voltage supplies, resistor and voltage tolerances, and the number of inputs per stage (m). The unknowns are the three resistors (R_1, R_2, R_T) and the number of outputs per stage (n). The solution will be optimized by minimizing the required base current ($\underline{I}_{B1}$). This will be done by assuming various values of $\underline{I}_{B1}$ and calculating the maximum number of outputs ($\bar{n}$) for each value of $\underline{I}_{B1}$. In this way a chart of $\bar{n}$ vs $\underline{I}_{B1}$ is generated. From this chart the minimum current corresponding to the desired value of n is selected and the final circuit is calculated. The procedure is illustrated with a sample design.

A transistor type is chosen so that the following become known: $\overline{I}_{CO}$, $\underline{B}$, $\underline{V}_{BU}$, $\overline{V}_{CU}$, and the worst base and collector characteristics. In addition, ρ, σ, V_{BB}, V_{CC}, and m are chosen. This leaves four unknowns: R_1, R_2, R_T, and n. Since there are four unknowns but only two basic equations, two intermediate variables are chosen in order to increase the number of available equations to equal the number of unknowns. The intermediate variables chosen are $\underline{I}_{B1}$ and V_{CD1}.

It is now possible to draw the worst case circuits in simplified sections and to write equations from them. Fig. 10(a)–10(c) describe the worst case for V_{CU}. Fig. 10(d) describes the worst case for V_{BU}.

Eqs. (24) to (27) are written from the four circuits.

$$\frac{V_{CD1} - V_{BD1}}{(1+\rho)R_T} = \frac{(1+\sigma)V_{BB} + V_{BD1}}{(1-\rho)R_2} + \frac{V_{BD1}(m-1)}{(1-\rho)R_T} + \underline{I}_{B1} \quad (24)$$

$$\frac{(1-\sigma)V_{CC} - V_{CD1}}{(1+\rho)R_1} = \frac{V_{CD1} - V_{BD1}}{(1+\rho)R_T} + \overline{I}_{CO} + \frac{V_{CD1}(n-1)}{(1-\rho)R_T} \quad (25)$$

$$\underline{I}_{C1} = \frac{(1-\sigma)V_{CC} - \overline{V}_{CU}}{(1-\rho)R_1} + \frac{(\overline{V}_{BD3} - \overline{V}_{CU})(n-1)}{(1-\rho)R_T} - \frac{\overline{V}_{CU} + V_{BU}}{(1+\rho)R_T} \quad (26)$$

$$\frac{(V_{CU} + \underline{V}_{BU})m}{(1-\rho)R_T} = \frac{(1-\sigma)V_{BB} - \underline{V}_{BU}}{(1+\rho)R_2} - \overline{I}_{C0}. \quad (27)$$

These equations correspond to (18), (23), (15), and (14). In order to facilitate computation in the preliminary steps of the solution, it is advisable to make two approximations:

$$V_{CU} \doteq \overline{V}_{CU}.$$

In Fig. 10(c): $\underline{I}_{C1} \doteq I_{11}$, so that (26) can be written:

$$\underline{I}_{C1} \doteq \frac{(1-\sigma)V_{CC} - \overline{V}_{CU}}{(1-\rho)R_1}. \quad (28)$$

Choosing the 2N135 for the sample design and assigning values for m, the voltages, tolerances, and letting $\underline{I}_{B1} = 0.4$ ma for the first trial, the following parameters are known:

$V_{CC} = 19$ volts; $V_{BB} = 10$ volts; $\sigma = 0.05$;

$\rho = 0.05$; $\overline{I}_{C0} = 0.05$ ma; $\underline{V}_{BU} = 0.15$ volt;

$\overline{V}_{CU} = 0.1$ volt; $m = 5$; $\underline{I}_{B1} = 0.4$ ma;

$\underline{I}_{C1} = 6.9$ ma (using a 0.9 safety factor); $V_{BD1} = 0.26$ volt.

Only the last three values are functions of $\underline{I}_{B1}$ and subject to change for succeeding trials. The other values remain fixed.

The values listed above are substituted into (24), (25), (27), (28); R_1, R_2, and R_T are eliminated to obtain the relation:

$$n = 14.48 - 0.682V_{CD1} - \frac{38.35}{V_{CD1}}. \quad (29)$$

To determine the maximum value of n,

$$\frac{\partial n}{\partial V_{CD1}} = 0 = -0.682 + \frac{38.35}{V_{CD1}^2}. \quad (30)$$

$V_{CD1} = 7.5$ volts, and $n = 4.2$. With $\underline{I}_{B1} = 0.4$ ma and $m = 5$, the maximum value of n is 4.2. The calculations are repeated with various values of $\underline{I}_{B1}$ to obtain Table I.

TABLE I

$m=5$						
$\underline{I}_{B1}$	0.8	0.4	0.3	0.25	0.2	0.1
$\bar{n}$	4.23	4.2	4.13	4.06	3.91	2.99
V_{CD1}	7.52	7.5	7.38	7.28	7.27	6.99

The value of n drops sharply as $\underline{I}_{B1}$ approaches $\overline{I}_{C0}$. The highest integral value of n which can be obtained is 4 and the lowest base current required is 0.25 ma.

For

$\underline{I}_{B1} = 0.25$ ma: $\underline{I}_{C1} = 4.38$ ma.

$V_{BD1} = 0.233$ volt.

$V_{CD1} = 7.28$ volts.

From (28)

$$R_1 = \frac{(1-\sigma)V_{CC} - \overline{V}_{CU}}{(1-\rho)\underline{I}_{C1}} = \frac{18 - 0.1}{(0.95)(4.38)} = 4.3 \text{ K}. \quad (31)$$

From (24) and (27), $R_2 = 58.55$ K; $R_T = 12.95$ K. With these resistance values the approximations made before can now be corrected. More accurate values can be calculated for V_{BU1}, V_{CU1}, and $\overline{V}_{BD3}$, and (26) can be used instead of (28). Using (16) to calculate V_{BU1}, $V_{BU1} = +0.195$ volt. In order to find V_{CU1}, it is first necessary to calculate I_{B1}. The only difference between I_{B1} and $\underline{I}_{B1}$ is due to the change in V_{BB}.

$$\Delta I_{B1} \doteq -\frac{1}{R_2}\Delta V_{BB}$$

$$I_{B1} \doteq \underline{I}_{B1} + \frac{\overline{V}_{BB} - \underline{V}_{BB}}{R_2} = 0.25 + \frac{1}{58.55} = 0.267 \text{ ma}. \quad (32)$$

From the worst collector characteristics:

At:

$$\left.\begin{aligned} I_B &= 0.267 \text{ ma} \\ I_C &= 4.86 \text{ ma} \end{aligned}\right\} V_{CU} = 0.084 \text{ volt}.$$

Using (21) to calculate $\overline{V}_{BD3}$:

$\overline{V}_{BD3} = 0.36$ volt. ($\overline{V}_{B3} = 0.26$ volt; $\overline{R}_{B3} = 0.04$ K).

Eqs. (24) through (27) are now solved, using all values corresponding to an $\underline{I}_{B1}$ of 0.25 ma, to obtain:

$R_1 = 4.35$ K; $R_2 = 62.3$ K; $R_T = 13.2$ K; $n = 4.09$.

The result obtained is a general purpose stage with $m = 5$ and $n = 4$. It has been optimized to use the minimum possible base current. If the logical designer requires a standard stage with a higher number of inputs and outputs, then a transistor with higher current gain is required.

Hand Solution of an Intermediate System

There is a need for logical systems composed of stages having various combinations of m and n. These stages must be compatible, *i.e.*, it must be possible to connect them to each other. Such circuits can be designed by adopting a standard value for V_{CD1}. The base circuit of each type of stage is determined by the number of inputs (m) and the number of outputs (n), *not* by the load which each output is required to drive. The maximum load which any following stage may present to the collector of the driving stage is defined (for the whole system) as the maximum allowable load current. This maximum load current may be required by *various*

mbinations of m and n, and any circuit which requires is maximum allowable current is referred to as a aximum load circuit.

It is advantageous to let the standard V_{CD1} equal the V_{CD1} of the general purpose circuit ($m=5$, $n=4$) de-gned above, and to let the maximum allowable load rrent equal I_{T3} of that circuit.

$$V_{CD1} = 7.28 \text{ volts}, \ I_{T\max} = 0.508 \text{ ma}.$$

sing the same values as in the general purpose solution r V_{CC}, V_{BB}, ρ, σ, $\bar{I}_{CO}$, $\underline{V}_{BU}$, $\overline{V}_{CU}$, the transistor charac-ristics, and assigning n, a maximum load circuit is signed as follows:

R_1 is calculated from (25).

$\underline{I}_{C1}$ is calculated from (26).

$\underline{I}_{B1}$ and V_{BD1} are obtained from the transistor characteristics.

From Fig. 10(a):

$$R_T = \frac{V_{CD1} - V_{BD1}}{(1+\rho)I_{T3}} \tag{33}$$

R_T is calculated.

R_2 and m are calculated from (24) and (27).

ollowing this procedure for $n=2$:

$R_1 = 9.0$ K; $\underline{I}_{C1} = 2.1$ ma;

$\underline{I}_{B1} = 0.125$ ma (with a 0.9 safety factor);

$V_{BD1} = 0.20$ volt; $R_T = 13.2$ K; $m = 8.2$ (use $m = 8$);

$R_2 = 41.2$ K.

his circuit is compatible with the $m=5$, $n=4$ circuit ad is to be used for every stage in the system where $m=8$ and $n=2$. Other maximum load circuits can be de-gned for $n=1$, $n=3$, and $n=5$. If a system consists tirely of various maximum load circuits, it can be as-mbled without any further calculations, for every age will drive its maximum load and V_{CD1} will be the andard value.

The next circuits considered are those which require ss drive than the maximum load circuits, *e.g.*, a stage ith $m=2$, $n=2$, or a stage with $m=1$, $n=4$. Here m is nown. R_1, $\underline{I}_{C1}$, and V_{BD1} are calculated as before. Eqs. 4) and (27) are solved for R_T and R_2. The following alues are obtained for $m=2$, $n=2$:

$$R_1 = 9.0 \text{ K}; \ R_T = 31 \text{ K}; \ R_2 = 132 \text{ K}.$$

he load which this circuit presents to the stages driv-g it is 0.218 ma, from (33). Wherever this circuit is sed in the system, the difference between 0.218 ma nd the maximum permissible load (0.508 ma) should e taken up by a collector load resistor to ground, in rder to maintain V_{CD1} at design value.

Similar circuits can be calculated for any combination f m and n up to those corresponding to maximum load rcuits.

Test Results

General purpose circuits were designed, using both the computer solution and the hand computation solution. For the same requirements and the same transistor the two solutions gave almost identical results. The circuit was constructed in its worst case condition, using limit values of all parameters and a "worst" acceptable transistor, in order to check the failure points. Circuit failure occurred exactly as predicted by the analysis.

A binary adder/subtracter circuit was built using the hand computation solution. Many special purpose control circuits have been built using this solution. They have been in operation for six months without failure.

Conclusions

General Conclusions

With the procedures discussed in this paper, reliable resistor-coupled transistor logical circuits can be designed and built. Furthermore, it has been shown that the circuits can be optimized according to certain criteria. The solution can be programmed on a small digital computer (E101). Circuits for the general purpose system and intermediate system were designed, taking worst case conditions into consideration, and with minimum required transistor base current ($\underline{I}_{B1}$) as a circuit design criterion. The validity and accuracy of the design procedures were proven by building and testing typical circuits.

Circuit Considerations

The design procedures discussed and the circuits tested produced much useful information concerning the effects of certain circuit parameters on the circuit solution.

It was found that certain intermediate variables (V_{CU1}, V_{BU1}, and $\overline{V}_{BD3}$) have second-order effects on the circuit solution. It was found that the collector load is essentially equal for the two worst cases, and therefore the transistor collector current remains essentially constant. It was found that the current through the transfer resistors (R_T) connected to the collector of a conducting transistor is negligible. These facts were used to simplify the circuit design procedures.

The effects on the circuit design of the variation of circuit parameters were studied by observing their influence on the minimum required base current ($\underline{I}_{B1}$), using a computer program. Doubling the value of the maximum transistor collector cutoff current ($\bar{I}_{CO}$) more than doubled $\underline{I}_{B1}$. Doubling the value of the minimum large signal current gain ($\underline{B}$) decreased $\underline{I}_{B1}$ by a factor of 5. Decreasing the power supply tolerance (σ) from ± 5 per cent to ± 2 per cent decreased $\underline{I}_{B1}$ by a small amount, however, decreasing the resistor tolerance (ρ) from ± 5 per cent to ± 2 per cent, as well as σ, caused $\underline{I}_{B1}$ to decrease by 50 per cent. A decrease of the number of inputs (m) or outputs (n) causes $\underline{I}_{B1}$ to

decrease, however (n) has the greater effect. Doubling the collector power supply voltage (V_{CC}) causes $\underline{I}_{B1}$ to decrease by about 50 per cent. A change in the base power supply voltage (V_{BB}) has little effect. Lowering the minimum required base voltage ($\underline{V}_{BU}$) lowers $\underline{I}_{B1}$. Raising the maximum collector voltage under saturated conditions ($\overline{V}_{CU}$) raises $\underline{I}_{B1}$. Those variations which cause a decrease in $\underline{I}_{B1}$ will have a favorable effect on other circuit considerations, *e.g.*, maximum number of circuit inputs (m), maximum number of circuit outputs (n).

Comparison of Computer and Hand Solutions

A comparison of the computer and hand solutions indicates that each method has certain advantages.

It is possible to calculate a solution to about the same accuracy using either method. The chances of making a computational error on a computer are remote. The transistor is approximated by equivalent circuits which give simple mathematical expressions for use in the computer solutions and will lead to some small inaccuracies in the calculated results. In a hand solution the transistor characteristics are used directly.

The time to complete an optimized hand solution for one set of circuit conditions, using the methods of this paper, was about four hours. The time to calculate the same solution on the E101 computer was about one hour. However, it took the equivalent of about six full working weeks to obtain the computer program. If it is desired to obtain a small number of circuit solutions, the hand solution methods are satisfactory. If it is desired to obtain a large number of solutions and to study the effects on these solutions of varying circuit parameters, then the investment of the necessary programming time has been worthwhile.

With a hand solution, the engineer does the computation and decision making. He therefore gains a better understanding of the circuit operation. The computer solution does the decision making as well as the computation.

It might be advantageous to place the hand solution on the computer, using the computer to do all the calculations and permitting the engineer to make the decisions.

Appendix I

The following definitions and symbols are used in this paper.

V_{BB} = base supply voltage.
V_{CC} = collector supply voltage.
$1/G_1 = R_1$ = collector load resistor.
$1/G_2 = R_2$ = resistor connecting a base to V_{BB}.
$1/G_T = R_T$ = transfer resistor connecting a collector to a succeeding base.
$V_{BU} = V_{\text{Base, Up}}$. Base voltage of a cutoff transistor.

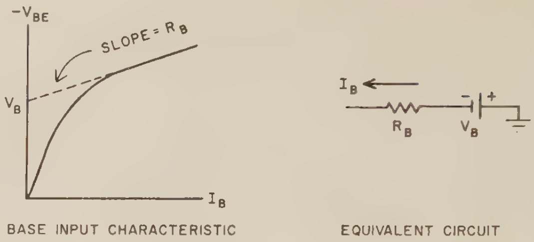


Fig. 11—Base input characteristic equivalent circuit.

$V_{CU} = V_{\text{Collector, Up}}$. Collector voltage of a conducting transistor.
m = number of inputs to a base.
n = number of outputs from a collector.
I_B = base current.
$V_{BD} = V_{\text{Base, Down}}$. Base voltage of a conducting transistor.
$V_{CD} = V_{\text{Collector, Down}}$. Collector voltage of a cutoff transistor.
I_C = collector current.
I_{CO} = cutoff collector current, emitter open circuited.
B = large signal current gain I_C/I_B.
ρ = resistor tolerance.
σ = voltage supply tolerance.

It will be convenient to represent the forward-biased base input characteristics of the grounded emitter configuration by an equivalent circuit (Fig. 11).

V_B = base input equivalent voltage.
R_B = base input equivalent resistance.

The symbol V stands for the magnitude of a voltage. Polarity is indicated by $+$ or $-$.

Appendix II

Calculation of V_{CU1}

$\overline{V}_{CU1}$ is the maximum collector voltage of the stage being designed. $\underline{I}_{B1}$ has been defined as the transistor base current flowing under worst conditions for $\overline{V}_{CU1}$. V_{CU1} is the collector voltage and I_{B1} is the base current which correspond to the worst case for $\underline{V}_{BU1}$. In Fig. 12 the two characteristics for $I_B=\underline{I}_{B1}$ and $I_B=I_{B1}$ are shown. The load line is shown as a horizontal line at $I_C=\underline{I}_{C1}=\underline{BI}_{B1}$ in the region of saturation because the collector currents flowing in the two worst cases are almost equal. The actual value of the collector voltage under worst conditions for $\underline{V}_{BU1}$ ($I_B=I_{B1}$) is V_{CU1*}(Fig. 12). The collector characteristic for $I_B=I_{B1}$ can be approximated in the region of saturation by a straight line between the origin and the intercept of the $I_B=I_{B1}$ curve with the $V_C=\overline{V}_{CU1}$ line. The approximation

$$V_{CU1} = \overline{V}_{CU1}\frac{\underline{BI}_{B1}}{\underline{B}I_{B1}} = \overline{V}_{CU1}\frac{\underline{I}_{B1}}{I_{B1}}$$

will give a value of $V_{CU1} > V_{CU1*}$ which is on the safe

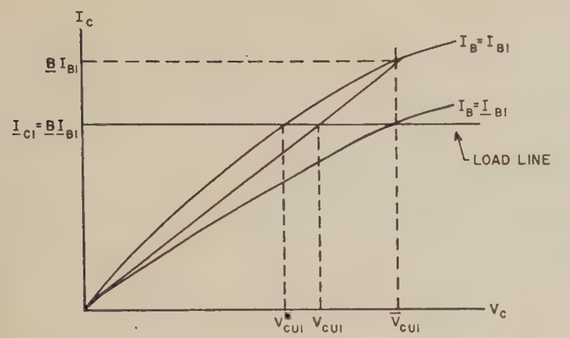


Fig. 12—Collector characteristics for calculation of V_{CU1}.

le. Since I_{B1} is very close to $\underline{I}_{B1}$, the straight line as-mption of the collector characteristic in the region of turation gives a good approximation of V_{CU1*}.

Appendix III

In the computer solution discussed, an iteration rou-e is used. A value is assumed for V_{CD1} and a circuit ution calculated. A value for V_{CD1}' is then calculated m the circuit solution. This process is repeated with ferent assumed values until $V_{CD1}' \doteq V_{CD1}$. The be-vior of this solution is described by a hyperbola (Fig. where V_{CD1}' is expressed as a function of the as-med value V_{CD1}. This equation can be obtained by solving (14), (15), and (18) simultaneously for G_1 and G_T as functions of V_{CD1}. The results are substituted in (23) to obtain V_{CD1}' (the calculated value) as a function of V_{CD1} (the assumed value). This equation takes the form of an equilateral hyperbola where the y asymptote is defined by

$$h \doteq (1-\sigma)V_{CC}, \tag{34}$$

the x asymptote is defined by

$$k \doteq m(V_{BD1} + \underline{V}_{BU1} + V_{CU1}) - \frac{nV_{CO}}{\underline{B}}\left\{\frac{\bar{I}_{CO1}}{\underline{I}_{B1}} + 1\right\}, \tag{35}$$

and the distance l from the vertex to the intersection of the asymptotes is defined by:

$$\frac{l^2}{2} \doteq (1-\sigma)V_{CC}\left\{\frac{nV_{CC}}{\underline{B}}\left(\frac{\bar{I}_{CO1}}{\underline{I}_{B1}} + 1\right) + 2m[\rho V_{BD1} + (2\rho + \sigma)(\underline{V}_{BU1} + V_{CU1})]\right\}. \tag{36}$$

Acknowledgment

The authors wish to acknowledge the guidance, help, and encouragement of Dr. Howard E. Tompkins, who suggested the basic circuit for this paper and served as faculty supervisor for the thesis on which this paper is based.

On the Analysis of Sequential Machines*

R. G. GILLESPIE† and D. D. AUFENKAMP‡

Summary—**In this paper we indicate briefly how the methods of algebraic solution of Markov chains with constant transition babilities apply to the analysis of sequential machines. Mealy's del of a sequential machine is assumed. A stochastic matrix is ociated with each such machine to provide a starting point for the lysis. "Closed" sets of states are then characterized, for example, the appropriate theorems about Markov chains. A technique is mulated for reducing the connection matrix of any sequential chine to a canonical form.**

THE methods of the algebraic solution of Markov chains with constant transition probabilities provide techniques for the analysis of the matrix representation of any sequential machine. Considering the formulation established in two previous papers[1,2] these techniques can be used to reduce the connection matrix of a sequential machine to a canonical form.

We associate with the state diagram of a sequential machine a matrix, $P = [p_{ij}]$, called the *probability transition matrix*, as follows. If there is at least one input taking the machine from state i to state j, we designate a real number $p_{ij}(0 < p_{ij} \leq 1)$ as the *transition probability*

* Manuscript received by the PGEC, March 21, 1958.
† Boeing Airplane Co., Seattle, Wash.
‡ Lockheed Missile Systems Div., Palo Alto, Calif. Formerly at d College, Portland, Ore.

[1] F. E. Hohn, S. Seshu, and D. D. Aufenkamp, "The theory of nets," IRE Trans. on Electronic Computers, vol. EC-6, pp. 154–161; September, 1957.

[2] D. D. Aufenkamp and F. E. Hohn, "Analysis of sequential machines," IRE Trans. on Electronic Computers, vol. EC-6, pp. 276–285; December, 1957.

from state i to state j. If there is no input connecting i to j, then $p_{ij}=0$. In the event of a terminal state, k, we assign $p_{kk}=1$ without essentially altering the structure of the machine.

In this paper we impose the restriction that

$$\sum_{j=1}^{q} p_{ij} = 1$$

(where q is the number of states). Under this restriction the probability transition matrix P becomes a stochastic matrix.[3,4] We also define an initial probability distribution $\{a_k\}$, where a_k is the probability that the machine is initially in state k. The stochastic matrix P, together with the initial distribution $\{a_k\}$, completely defines a finite Markov chain with constant transition probabilities.[5] A sequential machine so described will be called a *Markov machine.*

If a transition from state j to state k in n steps can occur via the sequence of states $j, j_1, j_2, \cdots, j_{n-1}, k$, then the probability that the machine passes through this path, if initially in state j, is $p_{jj_1}p_{j_1j_2}\cdots p_{j_{n-1}k}$. Hence, the sum of the probabilities over all the distinct paths between states i and j is the probability of finding the machine in state k after a sequence of n inputs if the machine is initially in state j. This probability, called a *higher transition probability*, is denoted by $p_{ij}^{(n)}$. We have, for example,

$$p_{jk}^{(1)} = p_{jk}, \qquad p_{jk}^{(n+1)} = \sum_{\nu=1}^{q} p_{j\nu}p_{\nu k}^{(n)}$$

The higher transition probabilities can be calculated by taking powers of the probability matrix P. For instance, $P^{n+1}=[p_{ij}^{(n+1)}]=PP^n$. If the initial probability of find-the machine in state j is a_j, then the unconditional probability of finding the machine in state k after a sequence of n inputs is given by:

$$a_k^{(n)} = \sum_{j=1}^{q} a_j p_{jk}^{(n)}.$$

The state k can be reached from state j if there exists some n such that $p_{jk}^{(n)}>0$. A set of states S will be called *closed* if no one step transition is possible from any state of S to any state outside of S, that is, if $p_{jk}=0$ whenever state j is in S and state k is not. We note that if state j is in S and if state k is not, then $p_{jk}^{(n)}=0$ for all n. If state j is in S, then the sum of $P_{j\nu}$ extended over all those ν for which state ν is in S in unity, *i.e.*,

$$\sum_{\nu} p_{j\nu} = 1.$$

[3] W. Feller, "Probability Theory and Its Applications," John Wiley and Sons, Inc., New York, N. Y., vol. 1, p. 309; 1950.

[4] It may be that the nature of the sequential machine is such that the branch weights of the state diagram are already transition probabilities. As long as the condition

$$\sum_{j=1}^{q} p_{ij} = 1$$

holds, the subsequent development is relevant.

[5] Feller, *op. cit.*, p. 309.

Thus, in the jth row of P^n the elements corresponding to states in S add to unity. Therefore, if in the matrix P all rows and columns corresponding to states outside of S are deleted, the remaining submatrix is again a stochastic matrix. Hence, closed sets of states may be studied independently of the remaining states. A closed set of states is called *irreducible* if it does not contain a subset of states which in itself forms a closed set.

Next, we apply the method of generating functions to determine the number of irreducible sets of states in a sequential machine.[6] Define for fixed j and k:

$$P_{jk}(s) = \sum_{n=1}^{\infty} p_{jk}^{(n)}s^{n-1}.$$

Multiplying by $sp_{\nu j}$ and summing over j obtains

$$s\sum_{j=1}^{q} p_{\nu j}P_{jk}(s) = P_{\nu k}(s) - p_{\nu k}, \tag{1}$$

where q is the number of states. For each fixed k there is a system of q nonhomogeneous linear equations in the q unknowns $P_{1k}(s), P_{2k}(s), \cdots, P_{qk}(s)$. The determinant of the system, $D(s)$, is thus a polynomial in s of degree less or equal to q, while the $P_{\nu k}(s)$ are rational functions of s with the common denominator $D(s)$. One of the roots of the equation $D(s)=0$ is $s=1$ since

$$s\left(\sum_{j=1}^{q} p_{\nu j}\right) - 1 = 0.$$

On the other hand, each closed set of states contributes a factor of $(s-1)$ to the determinant, $D(s)$, of a system of equations formed as above. This assertion is evident from the following observation. Suppose the system of equations in the unknowns $P_{ik}(s)$ were rearranged such that the coefficient matrix had the form

$$\begin{bmatrix} A & 0\cdots 0 & 0 \\ 0 & B\cdots 0 & 0 \\ \cdot & \cdot\;\cdot\;\cdot\;\cdot & \cdot \\ 0 & 0\cdot \quad Y & 0 \\ \alpha & \beta\cdots\lambda & Z \end{bmatrix}$$

where closed sets of states are indicated by each of the square submatrices, $A, B, \cdots, Y$, while Z, although a square matrix, does not represent a closed set of states. The $\alpha, \beta, \cdots, \lambda$, and 0's are rectangular submatrices, and, in particular, the 0's represent null matrices. The determinant of the above matrix is $|A||B| \cdots |Y||Z|$. Since A represents a closed set of states, there is at least one factor of $(s-1)$ in $|A|$. If A represents more than one closed set of states, then there must be more than one factor of $(s-1)$ in $|A|$. In fact, if A represented, say, two irreducible sets of states, then rewriting A in the form

[6] Let $a_0, a_1, \cdots$ be a sequence of real numbers. If $A(s)=a_0+a_1s+a_2s^2+\cdots$ converges in some interval $-s_0<s<s_0$, then the function $A(s)$ is called the generating function of the sequence $\{a_j\}$. Feller, *op. cit.*, page 212.

$$A' = \begin{pmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ \alpha' & \beta' & Z' \end{pmatrix}$$

here A_1 and A_2 represent the irreducible sets, we see at $(s-1)$ must be a double factor of $|A|$. We reason milarly for $|B|, \cdots, |Y|$. The determinant of Z, wever, can not have a factor $(s-1)$ since Z does not present a set of states, *i.e.*,

$$s\left(\sum_j p_{ij}\right) - 1 \neq 0$$

r $s=1$, where the summation is over those states presented by Z. Hence the theorem:

Theorem I: The multiplicity of the root, $s=1$, in $D(s)=0$ indicates the number of irreducible sets of states in a sequential machine.

In the event that the equation $D(s)=0$ does not ve any multiple roots the following procedure can be ed to determine the $p_{jk}^{(n)}$ from which the structure of e machine can be determined. The technique is out-ed only; the reader is referred to Feller for its deriva-n.[7] We note that since the $P_{vk}(s)$ are rational it fol-ws that there exist coefficients, $\sigma_{vk}^{(1)}, \sigma_{vk}^{(2)}, \cdots, \sigma_{vk}^{(q)}$, ch that

$$P_{vk}^{(n)} = \frac{\sigma_{vk}^{(1)}}{s_1^n} + \frac{\sigma_{vk}^{(2)}}{s_2^n} + \cdots + \frac{\sigma_{vk}^{(q)}}{s_q^n} \tag{2}$$

ere $s_1, s_2, \cdots, s_q$ are the roots of $D(s)=0$. Now, con-der the two systems of equations

$$x_j = s \sum_{\nu=1}^{q} p_{j\nu} x_\nu \qquad (j = 1, 2, \cdots, q), \tag{3}$$

$$y_m = s \sum_{k=1}^{q} y_k p_{km} \qquad (m = 1, 2, \cdots, q). \tag{4}$$

e two systems have a common determinant, and ey have nontrivial solutions only for values of s for ich the determinant vanishes. Observe that the ove determinant equals the determinant, $D(s)$, of e (1). Let $s_1, s_2, \cdots, s_q$ be the roots (there are at ost q) of $D(s)=0$. For each s_r, $r=1, 2, \cdots, q$, the lutions $\{x_1^{(r)}, \cdots, x_q^{(r)}\}$ and $\{y_1^{(r)}, \cdots, y_q^{(r)}\}$ are termined to an artibrary multiplicative constant. The $^{(r)}$ in (2) are obtained from the relation

$$\sigma_{jk}^{(r)} = c_r x_j^{(r)} y_k^{(r)}, \tag{5}$$

ere the constants c_r are determined from[8]

$$1 = c_r \sum_{\nu=1}^{q} x_\nu^{(r)} y_\nu^{(r)} \tag{6}$$

[7] Feller, *op. cit.*, ch. 16.

[8] Although the solutions $x_\nu^{(r)}$ and $y_\nu^{(r)}$ are determined only to a merical factor, if we replace $x_j^{(r)}$ by $\alpha x_j^{(r)}$ and the $y_k^{(r)}$ by $\beta y_k^{(r)}$, en c_r will be changed into $c_r/\alpha\beta$ and consequently the $\sigma_{jk}^{(r)}$ in (5) l remain unchanged.

As an example, consider the following probability transition matrix:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

The systems of equations indicated by (3) and (4) are

$$\begin{aligned} x_1 &= \frac{sx_1}{2} + \frac{sx_2}{2} & y_1 &= \frac{sy_1}{2} + sy_2 + sy_4 \\ x_2 &= sx_1 & y_2 &= \frac{sy_1}{2} \\ x_3 &= sx_4 & y_3 &= 0 \\ x_4 &= sx_1 & y_4 &= sy_3 \end{aligned}$$

Since the multiplicative constant factor is arbitrary, we set $x_4=1$, $y_2=1$, and solve the resulting systems to obtain:

$$x_1 = \frac{1}{s}, \quad x_2 = \frac{2}{s^2} - \frac{1}{s}, \quad x_3 = s, \quad y_4 = \frac{2}{s^2} - \frac{1}{s},$$

and

$$y_1 = \frac{2}{s}, \quad y_2 = \frac{2}{s^2} - \frac{1}{s}, \quad y_3 = y_4 = 0.$$

The roots of $D(s)=0$ are $s_1=1$ and $s_2=-2$. Thus

$$\{x_j^{(1)}\} = \{1, 1, 1, 1\}, \quad \{x_j^{(2)}\} = \{-\tfrac{1}{2}, 1, -2, 1\},$$

$$\{y_m^{(1)}\} = \{2, 1, 0, 0\}, \text{ and } \{y_m^{(2)}\} = \{-1, 1, 0, 0\}.$$

From (6), the c_r are obtained: $c_1=\frac{1}{3}$, $c_2=\frac{2}{3}$. Calculating the $\sigma_{jk}^{(r)}$ from (5) and substituting these values in (2) we see that $p_{13}^{(n)}=p_{14}^{(n)}=p_{23}^{(n)}=p_{24}^{(n)}=0$, while the remaining $p_{jk}^{(n)}$ are different from zero. Therefore states 1 and 2 form a closed set of states which is evident from the state diagram in this example.

For machines in which more than one closed set of states exist the following method of analysis has been developed. Given the probability transition matrix $P = [p_{ij}]$ of a Markov machine having q states, consider the system of homogeneous equations

$$x_k = \sum_{j=1}^{k} p_{kj} x_j, \qquad k = 1, 2, \cdots, q.$$

This system will have a nontrivial solution if its determinant is zero. However, we know that a nontrivial solution exists, namely, $x_i=x_j$, $j=1, 2, \cdots, q$. Hence, the following theorem:

Theorem II: A necessary and sufficient condition that a set of states, $\{i_1, i_2, \cdots, i_t\}$, be a closed set is that in the system of homogeneous equations,

$$x_k = \sum_{j=1}^{q} p_{kj} x_j,$$

there exists a subset of t equations in the t unknowns, $x_{i_1}, x_{i_2}, \cdots x_{i_t}$.

Therefore, we now have a method of determining closed sets of states within closed sets of states, obtaining eventually the irreducible sets of states. In applying the above procedure the work is simplified if we group those equations whose right-hand member consists of two terms, etc. Examination of the system for subsets of k equations in k unknowns will yield all the closed sets of states.

As an example, consider the probability transition matrix:

$$\begin{array}{r} 1) \\ 2) \\ 3) \\ 4) \\ 5) \\ 6) \\ 7) \\ 8) \\ 9) \\ 10) \end{array} \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 \\ \frac{1}{5} & 0 & \frac{2}{5} & 0 & 0 & 0 & \frac{2}{5} & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{2}{3} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 \end{pmatrix}$$

Forming the equations

$$x_k = \sum_{j=1}^{q} p_{kj} x_j$$

and rearranging them in accordance with the above suggestion, we have

$$x_6 = 0$$
$$x_4 = x_4$$
$$x_7 = \tfrac{1}{2}x_2 + \tfrac{1}{2}x_6$$
$$x_5 = \tfrac{1}{3}x_1 + \tfrac{2}{3}x_5$$
$$x_8 = \tfrac{1}{3}x_1 + \tfrac{2}{3}x_5$$
$$x_1 = \tfrac{1}{2}x_1 + \tfrac{1}{4}x_5 + \tfrac{1}{4}x_8$$
$$x_2 = \tfrac{1}{5}x_1 + \tfrac{2}{5}x_3 + \tfrac{2}{5}x_7$$
$$x_3 = \tfrac{1}{4}x_1 + \tfrac{1}{4}x_7 + \tfrac{1}{2}x_8$$
$$x_9 = \tfrac{1}{6}x_1 + \tfrac{1}{6}x_3 + \tfrac{1}{6}x_6 + \tfrac{1}{2}x_9$$
$$x_{10} = \tfrac{1}{2}x_1 + \tfrac{1}{6}x_4 + \tfrac{1}{6}x_6 + \tfrac{1}{6}x_8.$$

States 4, 6, 5, 8 and 1 form a closed set of states. Decomposing this set we see that states 4 and 6 each form an irreducible set of states as does the set consisting of states 5, 8, and 1.

Theorems I and II permit the reduction of the probability transition matrix of any sequential machine to the form

$$\begin{pmatrix} A & 0 \cdots 0 & 0 \\ 0 & B \cdots 0 & 0 \\ \cdot & \cdot\ \cdot\ \cdot\ \cdot & \cdot \\ 0 & 0 \cdots Y & 0 \\ \alpha & \beta \cdots \lambda & Z \end{pmatrix}$$

where $A, B, \cdots, Y$ are square submatrices representing closed sets of states. Z is a square submatrix not necessarily representing a closed set of states. The $\alpha, \beta, \cdots, \lambda$ are rectangular arrays of entries which are all zeros if and only if Z represents a closed set of states. Each of the $A, B, \cdots$ may perhaps be further reduced into matrices of similar form

$$\begin{pmatrix} A' & 0 \cdots 0 & 0 \\ 0 & B' \cdots 0 & 0 \\ \cdot & \cdot\ \cdot\ \cdot\ \cdot & \cdot \\ 0 & 0 \cdots Y' & 0 \\ \alpha' & \beta' \cdots \lambda' & Z' \end{pmatrix}$$

and so on, until we arrive at the irreducible sets of states and the resulting canonical decomposition.

The preceding development serves to indicate very briefly the powerful methods for analysis of sequential machines provided by the techniques developed in studies of Markov chains.[9]

[9] A more complete study of these applications appears in the unpublished thesis of R. G. Gillespie presented to the Division of Mathematics and Natural Science, Reed College, in partial fulfillment of the requirements for the Degree of Bachelor of Arts.

Correction

Joseph O. Campeau, author of "The Synthesis and Analysis of Digital Systems by Boolean Matrices," which appeared on pages 231–241 of the December, 1957 issue of these TRANSACTIONS, has requested that the following corrections be made to his paper.

On page 238, in paragraphs two and four, the term $2^{k_i - 1}$ should be $2^{k_i} - 1$.

Immediately after (21), the following statement should be added: where the $\hat{a}_{ij}$'s are the values a particular input $\hat{a}_i$ can take.

The ranges on i and j in (22) should be

$$i = 1, 2, \cdots, p$$

$$j = 1, 2, \cdots, 2^{\left[\sum_{i=1}^{p} k_i\right]}.$$

Digital Computers in Continuous Control Systems*

EDWARD L. BRAUN†

Summary—**The use of digital computers in continuous control ems is discussed. A comparison is made of the two major types igital machines, namely the GP (general purpose) and the DDA ital differential analyzer), and characteristic features of each are sidered. Certain advantages and limitations of each type are cribed, together with their indicated areas of application. Methods rocessing input data from analog and digital sensing elements are cribed, as well as means of supplying control signals to output os.**

Advantages of Digital Computers

ANALOG and digital computers, separately or in combination, may be used in automatic control systems. Where applicable, a digital computer ers the following advantages.

) As much precision as may be required in any particular application.

) Improved reliability because of relative freedom from problems encountered in electromechanical or electronic analog computers, namely precision of components, wear, and stability.

) Relative insensitivity to mechanical or electrical "noise," which limits the complexity of problems soluble on an analog computer.

) Facilities for storage of large quantities of data that may be required in complex systems.

) The ability to perform logical as well as linear or nonlinear mathematical operations. This is a common requirement in applications calling for a master control to supervise an entire system.

) Flexibility derived from its logical capacity and the fact that its operations are controlled by a program rather than specially designed or interconnected physical components.

Review of Characteristics of Present GP and DDA Machines

The flow of information through a digital computer a control system is shown in Fig. 1. The chief differces in the two types of machines are related to the t that a DDA has a semifixed program, whereas that the GP is not fixed at all. Also, the arithmetic and itrol circuitry in a DDA are extremely simple.

The GP computer solves problems in a manner simi- to that used by an operator and desk calculator. The sic operation is to obtain two pieces of data from rage, perform a required operation on them, and nsfer the result back into storage, or use it as an

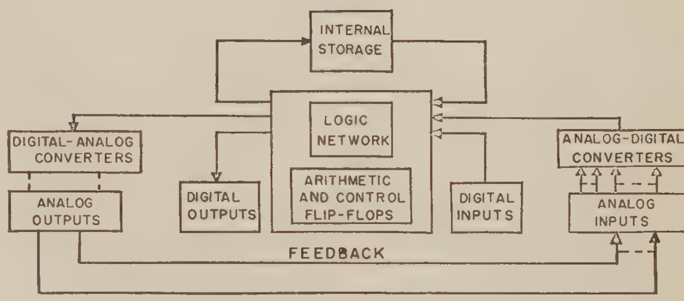


Fig. 1—Information flow through a digital computer in a control system.

output command. Whole numbers are transferred between the operational units, and the machine may accordingly be referred to as being of the absolute type.

The number of components external to the main storage is usually large, since control of the arithmetic unit must provide adequate sequencing and programming logic. Also, switching circuitry must be provided to connect the arithmetic unit to any part of the storage.

In a GP machine, time is consumed for search operations, *i.e.*, obtaining access to data and instructions in storage. Also, in many arithmetic operations, a considerable amount of redundant computing may be done if the values of the two items of information change very little from one computing cycle to the next. This is partially offset by the fact that in a GP machine, computing may be done in large increments, and therefore, the sampling rate can be reduced to the minimum value consistent with stability of a control system.

To solve a specific problem, not only initial values, but a detailed sequence of instructions for solving the problem, *i.e.*, a program, must be entered into the main storage unit.

The DDA[1] operates through an arrangement of switching and storage elements whose behavior is analogous to that of a system governed by the equations being solved. The information transferred between operational units in a DDA consists of single increments of the variables. Therefore, this type of machine is often referred to as an "incremental analyzer."

The operational steps required in a differential analyzer are permanently wired into the logic network. As a result, paired words comprising the "Y" and "R" registers of a digital integrator are read from the internal store. Because these words are permanently paired, and each pair is always operated upon in sequence, no time is consumed for "search" operations.

* Presented at Symposium on Computers in Simulation, Data luction, and Control, New York, N. Y., March 21, 1957. In Mr. un's absence the paper was read by Dr. Harold K. Skramstad, tional Bureau of Standards.

† Genesys Corp. (subsidiary of Chance-Vought Aircraft Co.,), Los Angeles, Calif.

[1] E. L. Braun, "Design features of current digital differential analyzers," 1954 IRE National Convention Record, pt. 4, pp. 87–97.

The arithmetic circuitry is simple for two reasons. First, the only arithmetic operation required is summation. Secondly, the summation is performed serially, a bit at a time. After each binary place of an integrator is operated upon, the result of that operation is sent to the magnetic drum for storage.

The control circuitry of the DDA has built into it a semifixed program. As a result, the solution of a problem requires only the following operations to be performed: 1) determination of how operational units are to be interconnected; this information is inserted into the machine by a simple code, 2) scaling of the problem, and 3) insertion of initial conditions into the drum storage.

It should be emphasized that the simplicity of the arithmetic and control circuitry in a DDA is not achieved by compromising performance, but is rather the result of the nature of the machine. The DDA is designed to simulate continuous systems by integration of a set of continuous variables starting from a set of prescribed initial conditions. The integration process is approximated to any desired degree of precision by means of a simple addition and storage process. All other mathematical operations that may be required, *e.g.*, multiplication, division, function generation, differentiation, can be derived for continuous functions, from this process.

For reference, the characteristics of the DDA which tend to simplify its control circuitry are listed below.

1) It has a semifixed program.
2) The only store to which access is required by each operational unit is the so-called *dz* store, which holds the current outputs of all the operational units. Its size is usually of the order of only a few hundred bits, depending basically on the number of operational units, and whether a binary or ternary representation of positive and negative values is used.
3) Only one bit at a time is read, recorded, or operated upon and always in a predetermined manner.
4) The arithmetic unit is comprised of simple adder circuits only.

The mode of operation of a GP machine requires that the control provide access to a main storage unit, holding several thousand words, and also that transmission links can be established between the main storage unit, the arithmetic unit, sequencing control, and input-output equipment, at the option of the programmer. Because of the great diversity of possible programs, a number of built-in transfer instructions must be made available to facilitate the task of programming.

If one were to generalize on the relative programming flexibility of GP and DDA machines, some features of the GP machine that could be cited are: its facility for using subroutines, the availability of more logical operations, better adaptability for the addition of new instructions, and the ease of programming access to any part of storage. The importance of programming flexibility depends, in part, on whether improved programs, and data on the control system's parameters are obtained, and, if so, when.

It is possible to build a GP computer with only one built-in arithmetic operation, namely subtraction, and to derive all other operations required during a computation by means of programming. However, this is not very useful, except possibly for purposes of teaching or amusement, because of the extremely cumbersome programming procedure that this entails. Also, it means an extremely inefficient use of storage capacity and a great reduction in speed of computation, compared to a GP machine having built-in arithmetic units for addition, multiplication, and division, as well as subtraction.

Areas of Application of GP and DDA Machines

The principal considerations that influence the design of a computer for continuous control system applications are given below.

The rate at which solutions have to be obtained. The response time of a digital computer in a control system is a function not only of the computer's speed, but also of the complexity of the system in which it is incorporated. In other words, since a digital computer functions by performing the steps of a computational routine serially the more steps there are, the longer it takes to perform a single cycle of computation.

The specified accuracy of the solutions. The subject of error analysis is outside the scope of this paper. However, a few comments are in order. Errors may arise from two sources (exclusive of equipment failures), those inherent in the nature of the computer and those peculiar to a specific problem.

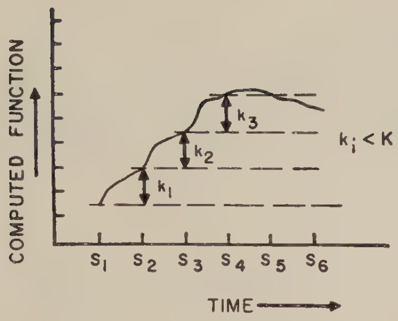
A source of error common to GP, DDA, and other machines is that due to round-off. The DDA is also subject to a special truncation error due to remainders in the R registers.

Another source of error is that of phase shift. This is caused by time lags in the transmission of data between parts of the system.

To minimize errors, the forms of the equations chosen must be such that they adequately represent the system's characteristics, and are also suited to the characteristics of the computer.

In a GP machine, the over-all error is governed largely by the numerical method and program selected. In the DDA, the general numerical method is relatively fixed, but there is still considerable choice both in the functions chosen and in the way they are generated. A characteristic of the DDA not shared by the GP machine is that an increase in the scale proportionately increases the time required for a solution, *i. e.*, product of the bandpass and reciprocal of the precision is a constant.

The form(s) in which information is (are) to be transmitted to various parts of the system. The principal differences between the GP and DDA machines in the internal transmission of information have been mentioned. Transmission from sensing elements and to actuators is discussed in the sections on input-output equipment.



. 2—Function generation possible with a GP or a DDA machine.

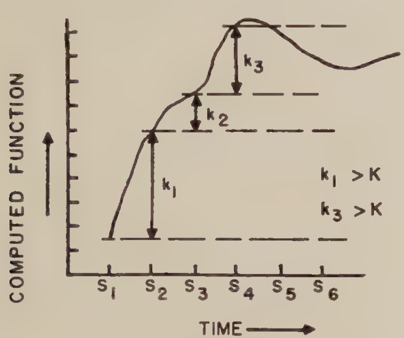


Fig. 3—Function generation requiring a GP machine.

The exact areas of application of DDA's and GP ma-
ines within the control system spectrum have not
en established to date. There are a number of reasons
this. First of all, there is the fact that this is still a
w field. Secondly, the characteristics of many of the
ntrol systems in which a digital computer may be in-
ıded have not been adequately specified. Third, there
he important point that a decision in respect to which
pe is better suited for a particular application cannot
made on the basis of the general nature of each ma-
ne alone. This decision is influenced to a large extent
the types of components available for fabricating
h machine. For example, before the advent of the
gnetic drum store, the utility of the DDA was some-
at questionable. Now, with new, compact, high-
ed, reliable components becoming available, it may
desirable to use these to replace the drum for applica-
ns calling for considerably higher speeds. However, if
ner components are used, *e.g.*, if the DDA's opera-
nal registers are fabricated from magnetic cores and
nsistors, its area of best application and the econom-
of its competitive position with GP or analog ma-
nes is shifted with respect to a DDA using a magnetic
ım. Accordingly, the statements made in the follow-
discussion concerning the relative advantages of
A and GP machines must be tempered by the under-
ng assumption that a DDA using a magnetic drum
re is being considered.

The DDA is the digital equivalent of an analog type
simulator, and suited for real time control systems
hin bounds set by speed requirements. If the registers
a DDA's operational units are stored on a magnetic

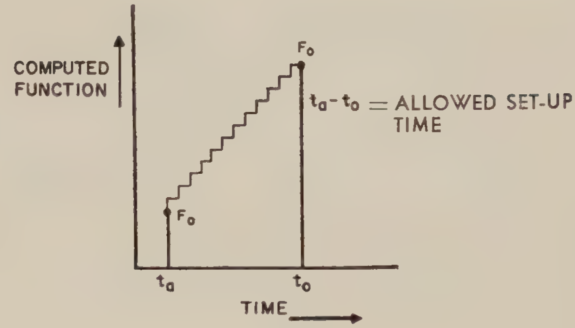


Fig. 4—Establishment of a computed initial condition in a DDA.

drum, the computing iteration rate is limited by the angular velocity of the drum. The pulse repetition rate is a function of the angular velocity and diameter of the drum, and the pulse packing density.

The reason why some computations can best be solved by a DDA and others by a GP machine can be seen by reference to Figs. 2 and 3. Fig. 2 shows successive values of a computed function at different sampling times. This function is such that the maximum change in its value between successive sampling times is always less than some value K. The number, K, represents the maximum number of increments that can be generated in a given time interval $(S_{i+1}-S_i)$ in a DDA. Therefore, if the value of the function to be computed does not change by more than K, in the time interval $(S_{i+1}-S_i)$, the use of a DDA is indicated because of its relatively greater efficiency in the computation of continuous functions. In Fig. 3, the computed function is such that the maximum change in its value between successive sampling times can be greater than K. Therefore, a DDA cannot be used.

Two other areas where a GP machine is indicated are: 1) where a large table of stored values must be consulted, and 2) where the interval of time allowed to establish correct initial conditions is too short for a DDA to do so.

The situation referred to by 2) is illustrated graphically in Fig. 4. At some time, t_a, an "alert" time, a signal is made available to the computer indicating that a new set of initial conditions is to be inserted into a computational loop. If the initial values of some of the variables must, in turn, be derived from a computational process, the following limitation must be considered. The DDA cannot generate more than some maximum number, say K, of increments during the interval Δt. Therefore, if during the allowed set up interval, Δt, the computed function must change by a number of increments greater than K, special means must be used to establish the correct conditions at time, t_0.

An Example of "Real-Time" Computation

An important application of digital computers is to provide guidance information automatically aboard high speed airborne vehicles. To achieve this purpose, the vehicle utilizes measuring and detecting instru-

ments which monitor its position in space and the environment, and controls which enable it to change its position in accordance with current requirements.

The computer receives its inputs from such sources as radars, gyro compasses, air speed indicators, roll and pitch indicators, barometric and radar altimeters, etc. It transforms this data to produce outputs in the form of control signals to the auto pilot, automatic tracking signals to the radars, display signals, etc. Derived quantities include present position in terms of latitude and longitude, distance to destination, etc.

Essentially, the guidance problem consists of resolving the aircraft's motion, as sensed by its instruments, along the axes of some chosen coordinate system. The computation problem is, therefore, basically that of solving trigonometric and algebraic equations, plus the integration of certain variables. The mathematical operations that will enter into the computation may be classified as follows:

- Addition and subtraction.
- Multiplication and division.
- Generation of trigonometric functions (which may be restricted to generation of sines, cosines).
- Generation of inverse trigonometric functions.
- Integration.
- Simple function generation: exponentials, absolute values, etc.
- Analog-to-digital conversion.
- Digital-to-analog conversion.

The computation may be performed either by an absolute or incremental computer. However, the nature of the problem particularly lends itself to an incremental technique, since the variables are, in general, continuous and have limited rates of change.

General Characteristics of the Input System

A control computer's function is to supply control signals to output servos. These commands are based on information received from sensing elements. Ideally, the information supplied by the sensors to the computer, and by the computer to the servos, should be in a form directly usable without further conversion. Generally speaking, this will not be the case.

An input sampling system for a digital computer in a control system may comprise, in general, the following elements.

1) Sensing devices (analog and/or digital).
2) Data converters (analog-to-analog, analog-to-digital, digital-to-digital).
3) Buffer storage.

The buffer performs the storage and switching functions required to transmit various encoded inputs to the computer in sequence. A timed sequence of orders to control these operations is supplied by the control unit of the computer.

These elements perform the following types of operations in synchronism with the computer: measuring, encoding, and time-division multiplexing.

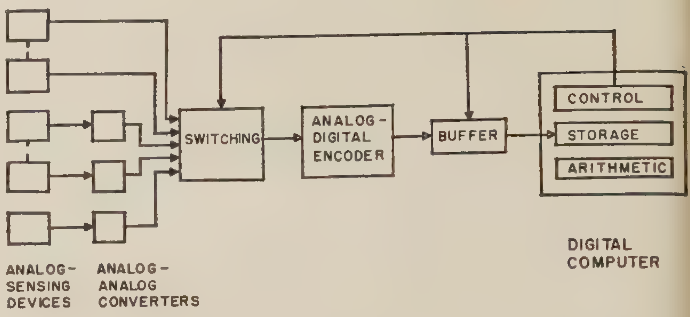


Fig. 5—An analog input system.

Fig. 5 illustrates a system for sampling a number of analog inputs in sequence. Two important means of equipment simplification are illustrated: time-sharing, and analog-to-analog converters.

To allow time-sharing of encoders, the outputs of instruments that are of the same form are combined by the use of sequencing switches controlled by signals from the control unit. For a large system, the cost of the sequencing switches and associated control equipments is outweighed by the simplification in encoding and buffering equipment.

By the use of analog-to-analog converters, the outputs of various instruments may be transformed to forms more amenable to encoding, and the time-sharing of encoding and buffering equipment increased. A simple example of an analog-to-analog converter is a potentiometer producing a voltage proportional to the position of a shaft to which it is attached.

Analog-to-Analog Converters

The number of possible representations of data which input instruments can produce is usually limited to four: 1) shaft position, 2) voltage, 3) time, and 4) frequency.

It is relatively simple to convert a time interval to binary representation. Therefore, one way of standardizing the input problem would be to convert shaft position and voltage to a time interval. There are two principal methods of voltage-encoding into the time domain. One method derives timing information by means of a generated sweep voltage and the other by means of a phase shifter. Fig. 6 shows block diagrams of each method.

In sweep-timing a direct voltage varies linearly from zero to a maximum expected value during each sampling period. When the unknown voltage equals the sweep voltage, the coincidence is detected. The time interval between the start of the ramp waveform, and the instant when it equals the unknown voltage is proportional to the value of the voltage.

In phase shift timing, the analog quantity is represented by the phase shift of a variable voltage with respect to a fixed reference voltage. The difference in phase is converted to a time interval by applying each

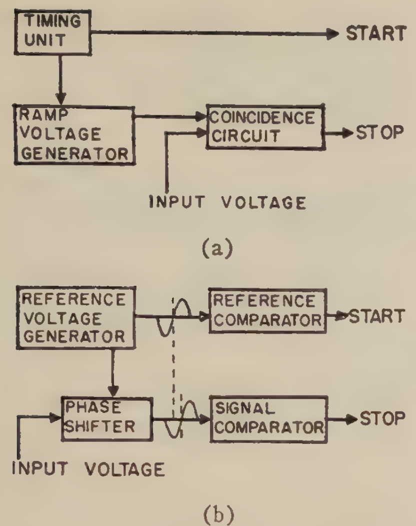


Fig. 6—Voltage-to-time encoders. (a) Generated sweep timing. (b) Phase shift timing.

ltage to a comparator. The interval is defined by the nes at which the reference and variable voltages, re- ectively, pass through 0°. This method is advanta- ous if the original analog quantity is easily converted a phase shift. It is used mainly to convert rotating aft data, since phase shift data can be obtained by eans of synchros.

The nonlinearity of the ramp voltage generator or the ase shifter and the uncertainty in the coincidence cuits limit the accuracy of the voltage-to-time do- in encoders.

Fig. 7 illustrates how a shaft position may be con- rted to a time interval by means of a resolver and a ase-shift type of voltage-to-time encoder.

alog-to-Digital Converters

Conversion of a Time-Interval to a Number: Under dis- ssion of analog-analog converters, means were shown converting a voltage to a time interval. Fig. 8 shows w a binary representation is obtained by counting the mber of clock pulses that appear in this interval. e required pulse repetition rate of the clock source pends on the specified precision and the input sam- ng rate.

Direct Conversion of a Shaft Position to a Binary Code: vices for converting shaft position to a binary code y be either of the incremental or absolute type. The remental type produces a pulse whenever the shaft ates an amount specified as a unit increment. In the solute type, the position of the shaft is monitored at times. The latter method provides the advantage of t being susceptible to an accumulation of error re- ting from lost pulses or "noise."

A widely used shaft position converter utilizes a led disk coupled to the shaft to be monitored. If ere are restrictions on the size of the disk, two or re disks may be driven in some suitably coupled ar- gement, with the less significant binary places being d from one disk and more significant places from the

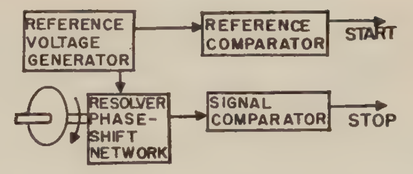


Fig. 7—Conversion of a shaft position to a time interval.

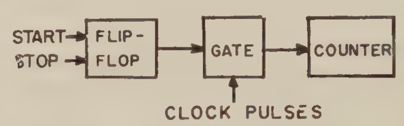


Fig. 8—Conversion of a time interval to a binary number.

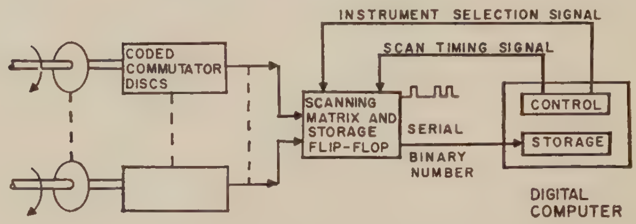


Fig. 9—A coded-disk type of angle encoder.

other. These disks may be read either optically or by brush contacts. They may be scanned either in a serial or parallel manner. To avoid ambiguities in a read-out, a reflected binary type of code or a special arrangement of brushes may be used. A schematic of a coded disk type of angle encoder is shown in Fig. 9.

There are a number of advantages to the coded-disk type of angle encoder:

1) Conversion of data from analog to digital form can be made without intermediate analog conversion steps.
2) The accuracy and repeatability of data using this system is relatively independent of variations in supply voltages, frequency, and temperature.
3) Maximum time-sharing is possible, and switching is performed on digital rather than analog signals.
4) The conversion time is compatible with the rate at which input data is sampled.

The Forms of Digital Inputs

Digital data can be obtained directly and/or from data converters in the following forms.

1) A whole number proportional to the value of an input variable, either serially, in the form of weighted pulse trains, or in parallel.
2) A number of pulses porportional to a change in the input.
3) A pulse rate proportional to the sampled value of the input.

Absolute measure has the advantage of not being susceptible to a permanent offset caused by a lost increment. Incremental measure may be satisfactory if it is inside a broader closed loop with zero steady-state error.

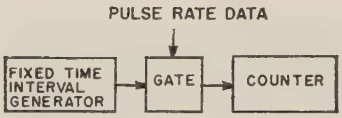


Fig. 10—Conversion of a frequency to a binary number.

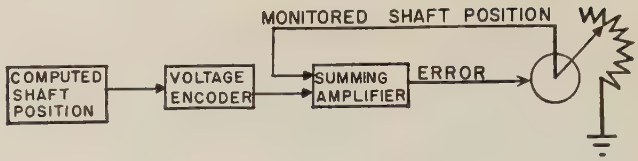


Fig. 11—Conversion of a number to a shaft position by a voltage comparison process.

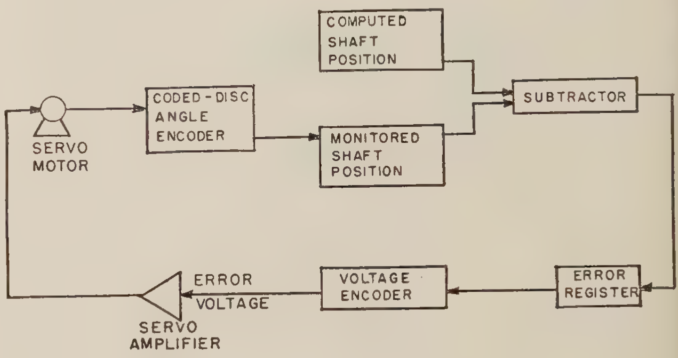


Fig. 12—Conversion of a number to a shaft position by a digital comparison process.

An example of a system where incremental measure may be satisfactory is one in which new absolute references are inserted at frequent intervals.

There is an important difference in the way input quantities are processed by a GP or a DDA. In a DDA, transmission of information between operational computing elements is by means of unitary weighted pulse trains. A special register, termed a servoregister, is used to generate a unitary weighted pulse train from the absolute binary input data. In the input problem, as well as within the computer, the characteristic limitation of the DDA, namely not being able to compute at a rate greater than K, presents itself. If the output of a servoregister is to be used as an independent variable input somewhere within the computer, then increments can be accepted from the servoregister only at a rate $\leq K$.

Digital-to-Digital Encoders

Conversion of a Frequency to a Number: Variable pulse rate data can be obtained from sources like pressure meters which produce a frequency proportional to pressure, or Doppler radars which emit a beat frequency proportional to the rate of change of range. This data can be converted to a binary number as in Fig. 10.

Output Signals to Control Elements

Data in absolute form, *i.e.*, as numbers, can be converted to control signals in the form of a voltage input to a servoamplifier. This requires a holding register and a voltage encoder. There is essentially no difference in the way this is done in a GP machine or a DDA.

Incremental signals, too, can be obtained from either a GP machine or a DDA. In the former case, increments are obtained simply by computing differences between successive control signals.

A method of converting a number to a shaft position, frequently employed formerly, is shown in Fig. 11. In this scheme, the positioning error is produced by a voltage comparison.

A system now used extensively is shown in Fig. 12. Here, the positioning error is produced by a digital comparison. The signal to the servoamplifier can be of the ON-OFF type or made proportional to the error over some specified range by use of an error register. The error register is set to the semiproportional digital difference of the monitored and desired shaft position. The conversion from the semiproportional digital error, contained in the error register, to error voltage is affected by using a binary weighted network. The output of this network is sent to a holding or desampling device which presents the output data to the controlled element as continuous, *i.e.*, smoothed data. The simplest method is to clamp the last sample until a new one is computed.

A relatively simple type of servodrive used extensively in aircraft control systems is the dc pulsing, or notched, motor. It produces an increment of rotation for each pulse input.

A voltage proportional to a pulse rate can be obtained simply by causing each pulse to add a fixed amount of voltage to a capacitative storage circuit.

Concluding Remarks

It is difficult to compare different classes of machines, especially when so many variations of design exist and so many more are possible within each class. Many items must be considered, *e.g.*, the type and complexity of the application, precision and speed requirements, choice of components. Many items which enter into an integrated system design are too detailed and variable to be treated here. However, it is not unusual for these details to vitiate arguments, based on general considerations alone, in favor of the superiority of one type of machine. For example, consider the problem of reliability. Often, reliability must be obtained at the cost of other factors. There are many devices that may be used in an attempt to improve reliability; proper selection and rating of components, duplication of equipment, duplication of programs, programmed checks, built-in checks, etc. Each of these affects the design of a particular machine, and may affect one design to a greater degree than another.

In short, conclusions about the aptness of a computer for a specific application cannot be reached without reference to specific computer embodiments, and the whole system of which the computer is a part. Computer "A" is better than Computer "B" only if the system including "A" more nearly satisfies specified requirements.

Computers in Process Industry Control*

WILLIAM F. GUNNING†

Summary—**Successful inclusion of a digital computer in the real-ne control loop of an industrial fluid process plant is dependent on practical steps: 1) the formulation of adequate mathematical odels, 2) the development of satisfactory data acquisition tech-ques, and 3) the proof of sufficient reliability. Examples are taken om the petroleum industry.**

Introduction

THIS paper considers some past, present and future applications of computers in simulation, data reduction, and control in the process industry. The rocess industry, by definition, is largely made up of ompanies whose raw materials, intermediates, and roducts are primarily fluids. While most of the examles used in this paper are taken from the petroleum inustry, the basic problems are found in plants producing ıch diversified items as paint, catsup, plastics, and ottled gas.

Over the past 20 to 30 years, the process industry has een a leader in the development and application of ervomechanisms for what might be called "minor op" industrial control. Many of the automatic regutors now in use contain ingenious, reliable, and inexensive analog devices for addition, differentiation, and tegration. Although they are seldom glamorized by e term "computer," it is difficult to distinguish beveen the information-processing they perform and that one by components of present day analog computers.

The ease of handling afforded by fluids, in combinaon with this simplified version of automatic control, as resulted in an industry that needs comparatively w workers per dollar of product produced. At first ance, it would hardly seem justifiable, therefore, to dd a hundred thousand dollar digital computer to ıch an installation. It most certainly cannot be justified n the basis of saving the money now spent on manual ontrol operations, as is the case in less automized aplications such as accounting and engineering design alculation.

Minor-loop process control has its limitations, howver, especially in relation to process efficiency, called yield," and in quality control. For example, present ay electronic regulators can hold process temperature ariations to within a few hundredths of a degree, yet, ven with such tight control on such a primary variable s temperature, it is often not possible to produce constently the exact product required. The complex facors leading to truly effective control generally cannot e measured directly, but must be determined by comining information from several primary measuring instruments. And this, then, is the justification for considering the use of major computing elements in process control.

The stored program digital computer was proposed as the logical candidate for this role at least five years ago. While it is true that both analog and digital computers are available today with speeds capable of real-time control, and while several significant steps have been made toward their application in such control, much additional work remains. The problems may be grouped roughly as follows:

1) The formulation and verification of satisfactory mathematical models.
2) Development of adequate data acquisition techniques and equipments.
3) The accumulation of convincing proof that the advanced computer and control equipment required is sufficiently reliable for process control.

The Mathematical Model in Plant Design and Operation

The chemical and petro-chemical industries are presently in the early stages of using both analog and digital computers as aids in 1) the design of new plants, and 2) in the operation of existing plants.

The basic equations of mass transfer, fluid dynamics, and chemical reaction rates are considered to be too complex to be used directly in the design—by classical methods—of new plants or processes. Equations take the form of nonlinear differential equations and partial differential equations that also include arbitrary functional relationships based on empirical data. As a result, the design of new plants has been based largely on extensions of existing working designs. Because of this empirical approach, designers have included substantial safety factors. Each new design represents only a small deviation from its predecessors.

On the other hand, the airframe industry has been virtually forced to pioneer in the use of computers in design. Since a contrivance built with the generous safety factors used in other industries would never get off the ground, these people have had to use computers to aid design, in order to have product at all. The process industry does not have the same compelling impetus to use computer technique but, nonetheless, has some important economic reasons for being interested. For example, if company A produces a plant which is more efficient in design, or produces a better product, then company B, addicted to the status quo and avoiding the risks of trying new techniques, may have a product, but no business.

* Presented at Symposium on Computers in Simulation, Data eduction, and Control, New York, N. Y., March 21, 1957.

† Beckman Instruments Co., Fullerton, Calif.

Automatic control in the process industry necessarily differs from other forms of "automation." The process industry has an advantage over "Detroit type automation" in that its materials are much easier to handle and control. On the other hand, it is much more difficult to define, in exact mathematical terms, the operations which are to be performed. Defining the problem of metal cutting is a matter of specifying where to place the work with respect to the cutting tool, and the rate at which the process will be carried out. The inter-actions in the production process in a chemical or petroleum plant are very complex, involve a considerable amount of cross-coupling and nonlinearity, and can be a mathematical trap for the casual promoter of automation.

Before we can hope to mechanize the decisions involved in controlling a refinery, we must be able to formulate, in mathematical-logical terms, the exact response that is to be made for each operating contingency. What is needed, then, is a reasonable mathematical model of the process to be automatized.

The level of the mathematical model can be quite varied. The following 5 classes that apply in the petroleum industry are listed in increasing order of generality or scope.

1) A model can be made of an *individual* component in a production process such as a distillation tower or a heat exchanger. An example will be given later.

2) A model can be formulated of a presently operating *process*, including its operating personnel. This model may avoid the formulation of detailed models of the first class. Such models are already being used to schedule efficient production operations in existing refineries. The model takes into account such factors as the storage facilities available, raw materials available, and various other economic factors, as well as descriptions of the plant yield characteristics. Manipulation of the model results in efficient operating instructions for use by the plant operators in setting automatic production control devices.

3) A model can include a *complete refinery* involving several plants or processes. Since the products of one process can be used as an input or as a fuel at another part of a refinery, the scheduling can be a very complicated affair. Texaco has used an IBM 705 to manipulate such a model to find the most economical and efficient balance in assignment and selection of various crudes and end products.

4) A more general mathematical model can include *company-wide* operations. This model would include such factors as: a) the scheduling of crude production, including a model of the company's oil reserves, b) crude transportation and shipping schedules, c) the refinery program as it relates to the division of assignments among several refineries, d) the assignment of storage and distribution facilities for the finished products, and e) market information based on past sales. The Standard Oil Company of Ohio is reported to be simultaneously simulating three refinery operations on an IBM 705 Computer.[1]

5) An ultimate extension of the scope of the model includes an *entire industry*. In addition to the factors mentioned before, the techniques of the theory of games will no doubt be used to explore the effects of various marketing and production strategies on an industry-wide basis, if, indeed, this is not already being done. (It might be strategically important to be quiet about such activities.)

As an example of a model of the first class, let us consider one of the basic tools used in chemical engineering, the multiple tray distillation tower. This device is very much like a many layered coffee percolator. The input material is fed into the center usually as a hot gas. The light components bubble up through the trays and eventually come out the top as a gas, while the heavier, less volatile components condense and drip down toward the bottom. Intermediate products may be extracted from intermediate trays. There are many other variations.

Many people have reported the results of using automatic digital computers to aid in the design of distillation towers. The first efforts appear to have used an IBM 602. A recent paper describes a typical calculation using a 650 and involving a 40 plate tower handling 15 fluid components.[2] In the latter, a set of temperatures and concentrations was determined in 10 to 15 minutes of computing time.

A mathematical model useful in designing a tower as such is not, however, necessarily best for investigating control operations. As has been true of missile guidance and control problems, analog simulation has been more helpful than digital simulation. Williams[3] reports a case which required the fairly large array of 84 amplifiers and 16 servos to handle a simplified 5-section, lumped model of such a tower. The liquid-vapor balance equations in each tray are nonlinear. The equations used in this simulation were simplified (although still nonlinear) in order to fit on the analog equipment available. Further, it was only possible to consider a two-component stream, whereas much more complex streams are typical. The analog computer approach, however, allows thousands of experiments to be made to explore the system behavior with assumed characteristics of various control schemes, thus approaching the techniques developed by the aircraft and guided missile control people. It is clear that analog simulation methods can operate in real-time and

[1] *Business Week*, April 7, 1956.

[2] Bonner, "Solution of Multicomponent Distillation Problems on a Stored Program Computer."

[3] *Indus. and Eng. Chem.*, vol. 48, p. 1008.

ow the actual physical controllers proposed for a tem to be connected into the simulated model to ify the part of the mathematical model used to deibe their action.

One should always worry about the validity of a thematical model, particularly if it has been necesy to make simplifying assumptions. Various methods ve been developed to take experimental data from its in production. The usual technique involves su-imposing a small sinusoidal variation as a "disbance" to a working piece of equipment and measur- its frequency response. The range of frequencies in ypical experiment of this type is from about 1/10 to cycles *per minute*. Time constants representing cors of the frequency response are in the range of 1 to 4 nutes. It is interesting to note that the instrumentan used in such a test is sensitive enough to get good in and phase data for temperature variation amplides down to ± 0.015°F.[4] With this sensitivity, it is ssible to conduct a frequency response experiment th such minute variations that the normal ink trace a production recorder-controller chart will not move ceptibly.

Measurements of gain and frequency response for ification of mathematical models involving flow ulation are plagued by noise or interference due to w turbulence and other acoustic disturbances. Satistory reduction of such data has, however, been ieved through the use of auto- and cross-correlation hniques.[5]

A large amount of work still has to be done in the eslishment of suitable mathematical models. Several portant devices used in chemical engineering have parently not been adequately described by models. cked towers loaded with china brick, coated with ious substances, is one example. A boiling heat exnger involving both fluid and vapor phase remains to adequately described.

The collection of information by automatic data logs (about which more is said in the following section), d the conversion of this information into a form actable to digital computers will facilitate the forman and verification of accurate mathematical models arger sections of the process by exploiting statistical hniques. A recent use of regression coefficient anals on an established process (not in petroleum) has closed that some 60 separate independent variables st be measured in order to predict the quality of a gle product with reasonable confidence. Cross-coung and nonlinear relationships need not be avoided in mulating mathematical models, as these may be nipulated by a computer to allow the behavior to compared with actual results measured with the data ging equipment.

A. R. Aikman, *J. Instr. Soc. Amer.*, vol. 3, no. 10, p. 412.
Reswich, *Control Eng.*, p. 50; June, 1955.

Data Acquisition

Most process industry plants are operated in a way that requires the logging of hundreds of variables. In many instances, the logs are prepared manually. Thus, log taking ordinarily represents an analog-to-digital conversion on the part of the man who writes down the numbers on the log sheet. The results are usually converted to units more meaningful to operating engineers and accountants. There is a growing trend in the process industry to use computer techniques, both analog and digital, to mechanize this operation. These automatic data loggers, as they are called, are usually independent of the recorders and controllers concerned with regulating the process.

Automatic data logging is made difficult by the variety of measuring instruments used, and the nature of the data that must be collected. To be economically feasible, data logging equipment must be multiplexed, or time shared, by a large number of different kinds of information sources. Since these sources vary widely in their sensitivity, or electrical response to a change in the measured phenomenon, a dynamic range of 100,000 to 1 is often required in the multiplexed equipment. For example, thermocouple temperature signals are an important class of process data, requiring high sensitivity. Ten microvolts per digit and sometimes even 1 microvolt per digit are typical values. The analog-to-digital conversion device must be flexible enough to handle the entire range.

Another problem is that signals collected from various parts of a plant that may be as large as a square mile in area cannot be counted upon to be referenced to a common ground potential. It is typical to find a volt or more of difference in "ground potential" between two different remote signal sources in such a situation. This means that, for consistent accuracy, the input signals must be measured by balanced or floating instruments independent of the "ground." Neither of the two input terminals for a channel may be common to any of the input terminals on any other channel.

Finally, it is important that data loggers have the ability to do some computing. The analog-to-digital conversion is of little value unless the digital printed log entries presented are in familiar and convenient units. This requires at least the following sorts of mathematical operation, all now available on present-day equipment:

1) Zero offset and scaling. This is the operation $Y = A + BX$.

2) Nonlinear function generation. Operations 1 and 2 suffice for converting thermocouple voltage measurements, for example, to temperature in degrees Fahrenheidt, Centigrade, or Rankin.

3) Square Root. This operation is necessary when flow is being measured as the square root of the pres-

sure drop measured across an orifice plate in a pipeline.

4) Multiplication and Division. To obtain consistent measurements of gaseous flow, it is necessary to "correct" for variations in temperature, pressure, and density.

5) Integration. This operation is essential if flow rate measurements are to be used in making a mass balance on flows in and out of a process or to calculate the amount of material moving into storage.

Integration is the most involved of these operations. Typically, many different integrals or totals must be maintained and regularly reported. Ordinarily, one hour totals as well as 24 hour totals must be calculated. Electrical analog methods have been used in data loggers. A pneumatic totalizer is available which combines the operations of integration and square root. Coincident current, magnetic-core storage is also used to keep running totals stored without drift.

An important development in process industry control is the concept of "stream analysis control," that is, the measurement of the actual chemical composition of the stream as it flows through the process. (The term "end point" analysis has also been used.)

In the past, process stream analysis has been conducted primarily on a sample basis by laboratory techniques, using wet chemistry or various types of spectrometric equipment. Because of the time lag, ranging from minutes to hours, stream analysis as such has had virtually no part to play in direct process control. This in turn has forced plant operators to run their processes by such "inferential" variables as temperature, pressure, or pH. (The inference is that if these are within bounds, the product stream is, too.) And this has held up the development of automatic control since the relationship between these inferential variables and the property one really wishes to control are complex and often obscure.

Certain laboratory-type analytical techniques are inherently fast enough to effect on-stream analysis, if computing capacity is available to analyze the record quickly. Such a device is the infrared spectrophotometer.

A typical record from an infrared spectrophotometer represents the "frequency response" of a mixture of materials; that is, the relative amount of light absorbed is plotted vs wavelength. One might guess from the complexity of detail of such a record that there is enough information to identify a large number of complicated components and, indeed, this is the case. However, the extraction of the desired information from such a record involves a considerable amount of computation. The Spectro-Sadic introduced about 5 years ago is an important specialized data acquisition device. It converts an essential part of the information, represented by a (mass) spectrograph chart, into digital language that is acceptable to a computer.

When spectrographic methods are used in stream analysis, the identities of the different stream components are usually known, but their relative amounts are unknown. In this case, the computational job is merely that of multiplying a vector by a matrix. Twenty to thirty components in the vector is typical. Such a problem, of course, can be handled and checked in a few seconds by most intermediate sized electronic digital computers. A matrix inversion is required for qualitative analysis, and this, of course, takes a little longer. However, in order for the information to be most useful, the computer must be available whenever it is required. Machines with the "program interrupt" feature can efficiently handle such spectrographic analysis and also be available for other computations involved in the control of a process or accounting work.

Spectrographic instruments are a powerful analytic tool, but their output represents a cross-coupled and inter-twined presentation of data in what is basically an inconvenient form. The use of a computer to make sense out of such data and introduce it into the real-time process control loop has served as a prime example of the way digital computers can be used. On the other hand, simpler, more effective continuous stream analysis methods (for example, nondispersive infrared and ultraviolet analysis or gas chromatography) are now becoming available. These eliminate the need for computation to determine the quantitative chemical composition. Freed of the data reduction task involved in spectrographic analysis, digital computers may bring their considerable talents to bear on the important job of computing control actions based on stream analyzer data, combined with other pressure, temperature, and flow data.

Present data logging speeds of 1 to 10 data samples per second are adequate for the alarm limit scanning and data logging functions for which the equipment has been designed. It seems likely that higher speed sampling will be required to allow the most to be made of sophisticated, computer-aided, statistically-oriented, investigations aimed at extracting the control relationships from test data taken on existing pilot or production operating units.

Real gains in plant operating economy are possible if production is freed from the constraint of steady-state operation. Variations in feed stream content, or a change from one product to another, will no doubt require higher information processing rates. Technically feasible, but costly means are available for increasing sampling speeds by several orders of magnitude.

System Reliability

One modern refinery in Southern California is a high-production operation spread over many acres. As is usual, the process is regulated and the operating data are presented by means of instruments in a centralized control room. One of these instruments recently failed in a very subtle manner. This "upset" took 5 days to

straighten out, and cost the company on the order of $\frac{1}{2}$ million dollars in lost production. This illustrates not only the need for more sophisticated process control in one of our most advanced industries, but also underlines the need for absolute reliability, day after day, in every element in the control system.

The mean-free-time between failures of the analog control equipment presently used is of the order of 6 months. The mean-free-time of large present-day, high-speed, general-purpose, electronic, digital, computing machinery using vacuum tubes, is of the order of one day. Industrial management has been justifiably reticent about conversion to the more sophistocated, but less reliable hardware, and of assigning to it the increased system responsibilities required by more complete automatic control.

Much is being done through the use of solid-state devices and other improved engineering techniques to extend the reliability of basic digital computing equipment. Much more needs to be done before such equipment can be considered as a serious contender for real-time control. However, the inertial properties of most controlled systems open several avenues to an "effective" system reliability, wherein the failure of a part of the system may be permitted so long as it has negligible consequence to the process.

Sophisticated error detection and correction techniques can be included since the basic information rates in the process being controlled are low when compared with the information processing rates available in electronic computers. Marginal checking techniques have proved to be helpful in predicting system failures caused by components that drift or change gradually. In order that such techniques may be used, it is necessary either to have two computers or to be able to allow the system to "coast" over marginal checking periods of a single computer. "Breathing spells" in the schedule of a digital computer can be obtained by using it primarily to *correlate* information and to produce instructions not for the process, but for analog servoregulators similar to those now used in the process industry. The process itself would continue to be regulated for minor variations and disturbances by the servoregulators, although between corrections by the digital system it would not necessarily be operating at optimum. During such intervals, the digital equipment could be marginally checked or minor repairs undertaken.

Such an analog-digital computing and controlling team will afford new opportunities for reliable operation and error detection. The analog equipment can easily be arranged to reject set-point instructions from the digital computer that are "unreasonable." The digital equipment would, of course, be signalled concerning any such rejection. Furthermore, failing or drifting instruments or controllers can be detected by proper programming and mathematical model formulation. The short-time response of a thermocouple to temperature variations is perfectly adequate to assure the stability of temperature control. The long-time drift of the voltage-to-temperature conversion relation of a thermocouple or the drift in the characteristics of the associated temperature regulator, will allow the controlled temperatures to shift slowly. Such a subtle shift will usually produce a change in product quality. Since stream-analysis information will be a part of the over-all control feedback loop, adjustment in the regulator control point can compensate for such thermocouple or regulating device drifts.

It appears that it is possible, therefore, to consider using a single digital computer in a control loop if 1) error detection methods, 2) error and failure prediction methods, and 3) modular design techniques are incorporated. This last feature allows rapid maintenance to be performed by inexperienced personnel while the process coasts on the last set of instructions generated by the computer. It is conceivable that the computing system could itself schedule the optimum time for maintenance by predicting periods of smooth operation when no change in raw materials or other disturbances are called for by the over-all program.

Conclusions

The process industry is in the early stages of developing and applying accurate mathematical models and computer techniques to aid in the design of new plants.

Real-time control of plants using digital computer techniques is being approached from two directions. Digital computer mechanized operations research methods are scheduling the operations of large systems. Data acquisition equipment is introducing digital techniques at the detailed minute-by-minute operating level. The middle ground of combining primary instrument data with economic data in real-time control is still largely unexplored. The pieces of the large picture are being fitted together like a jig-saw puzzle. Many of the economically important pieces are already in place.

A combined computer control system that 1) uses the best of *today's* analog and digital building blocks, and 2) takes advantage of available redundancies, can achieve an operating reliability greater than that possible using either technique alone. Such a system will allow a practical realization of complete, real-time computer control of an important part of our productive economy.

Aspects of Real-Time Simulation*

WALTER F. BAUER†

Summary—**Present day digital computers are too slow for the comprehensive real-time simulation of complex electronic systems such as those involving guided missiles. Computers appearing 2–3 years hence will be sufficiently fast for these applications. There is and will continue to be fertile application fields for analog-digital computer combinations. An existing large-scale (Univac Scientific 1103A-Epsco converter—Electronic Associates) analog-digital system is described. System design and specifications are discussed which relate to real-time speeds, necessary accuracies, digital computer programming, and synchronizing the digital computer with the remaining system components. Techniques for exploiting each computer's advantages are mentioned. Applications in the guided missile field and in other fields are discussed.**

APPROXIMATELY two and one-half years ago the computing groups at the Ramo-Wooldridge Corporation made the decision to establish a combination analog-digital computer system for the real-time simulation of ballistic missile systems. The plans and specifications subsequently made will come to fruition next month with the delivery of a high-speed, high-capacity analog-digital converter which will effect the tie-in of a large-scale digital computer and a large-scale analog computer. In the following I would like to describe this system to you and comment on three additional items related to this large-scale system: techniques for using such a large-scale analog-digital system; advances in digital computer systems which will make them competitive with analog computers and analog-digital combinations for simulation; and the future of analog-digital combinations.

The comprehensive simulation of a ballistic missile system implies the precise handling of many low-frequency quantities, as well as the handling and integration of many quantities varying at high frequency. Furthermore, certain aspects of the system are inherently of a sampled data nature and involve discrete data, while other parts of the system are inherently of a continuous nature. Also, the requirement exists that all computations proceed at extremely rapid rates in view of the need to make tractable the simulation of a large number of "flights." The requirement is, of course, a direct consequence of the fact that a number of measured quantities are degraded by noise and a sampling must be made to determine the statistics of the random variables of the flight under various conditions.

The Univac Scientific 1103A computer which has been performing much of the simulation is too slow to perform the entire set of computations of the extended simulation model at the desired rate, that is, a real-time rate. The large-scale Electronic Associates analog installation, on the other hand, is too inaccurate to provide the precision necessary. Thus, despite the fact that these computers represent nearly the ultimate in commercially available computers in their respective classes, a large-scale analog-digital converter was ordered from the Epsco Company to provide a hybrid system with the desirable speed and precision.

The characteristics of the Univac Scientific Model 1103A computer and the 300-amplifier Electronic Associates computer installation are not emphasized here, for their characteristics can be found elsewhere. The converter equipment has 15 channels of analog-to-digital conversion and 15 channels of digital-to-analog conversion capacity. Ninety microseconds are required for conversion of all 15 channels from analog to digital quantities. The accuracy of the equipment will be 10 bits on a dynamic range of 18 bits which implies a 0.1 per cent or 1 millivolt accuracy, whichever is larger. Data to and from the channels can be presented or received at the 1103A computer under digital computer control with flexibility as to the channels chosen. Conversions can be made upon command of the digital computer or upon a signal received from an external clock source. Specifications for the conversion equipment were written to allow the data transfer between the 1103A and the converter to be done quickly and with a minimum of digital computer programming inconvenience. The characteristics of the control and some of the techniques for programming, as well as a description of the system, have been recorded in a paper.[1]

One of the design features of the conversion equipment will exploit the "program-interrupt" feature of the 1103A computer. This feature allows a signal from an external source to interrupt the computations. In usual operation, an external clock signals the conversion of all analog-digital channels and all digital-analog channels, and simultaneously interrupts the 1103A computer to "inform" it that new data is now available as a result of the conversions. With this technique the programmer, in laying out the computations, need pay attention only to the logic of the program and not to the timing of operations; in other words, the computations in the digital computer can proceed approximately asynchronous to the conversion frequency and the tedious counting and time estimation on the part of the digital computer programmer is unnecessary.

As to the techniques of use of this system, certain loops of the simulation which do not have a profound effect on those variables, which need to be measured

* Presented at Symposium on Computers in Simulation, Data Reduction, and Control, New York, N. Y., March 21, 1957.

† Ramo-Wooldridge Corp., Los Angeles, Calif.

[1] W. F. Bauer and G. P. West, "A system for general purpose analog-digital computation," *J. Assoc. Computing Machinery*, vol. 4, 1957.

precisely, are handled in the analog computer. The loops where precision is important are handled in the digital computer. In some cases the integration of certain variables is actually accomplished by using both the digital and analog computers. Recently[2] some of these techniques have been presented. The technique of using the analog and digital computers together consists essentially in separating the signal into low- and high-frequency components and handling the low-frequency components digitally, and the high-frequency components analog-wise. The ideas here are new and have neither been explored nor exploited to any significant extent, although the general technique of separating a signal into high- and low-frequency components is not new.

A number of recent advances in digital computers enhance the digital computer as an instrument for real-time simulation, control, and data reduction. One of these is the program-interrupt feature mentioned above which, to my knowledge, was first placed on the 1103 computer installed at the N.A.C.A. installation in Cleveland described by Turner and Rawlings.[3] This feature will undoubtedly be seen more and more in digital computers for all of these real-time purposes, for so frequently input to the computer occurs at an unpredictable time or in asynchronism to the computer operation. Another improvement beginning to be seen in digital computers is the time clock which can be interrogated under program control. This time clock can be a digital clock inside the digital computer, or an "analog" clock which must be sampled by the digital computer by means of an analog-digital converter. The first digital clock installed on a computer was that installed on the Whirlwind computer at the Massachusetts Institute of Technology. A third major area of improvement is the increased input-output speed allowed by the buffering systems, or the simultaneous compute-while-transfer operation which will appear first in a commercial model in the IBM 709 and Univac Scientific 1105 computers next year. The increased high-speed storage capacity is likewise important, for in the digital computer storage can always be traded for time.

The speed of the computer is perhaps the biggest item for real-time control. I recently heard of a large analog installation performing a real-time simulation of a complex system in approximately one minute per run. This simulation is probably the most complex ever attempted in this country. Because a digital check was necessary, the computations were completely programmed for a drum computer. As I recall, it required 20-odd hours to perform one simulation run. Considering a factor of 50 as the difference in speed between this drum computer and a computer of the 1103A or 704 variety, these latter computers could perform the computations in about 25 minutes. The Univac-LARC computer, which will appear in late 1958, being about ten times as fast as the 1103A or 704, would perform the computations in about two and one-half minutes. The IBM STRETCH computer, which will probably appear in 1960, will be able to perform the computations for one simulation run in approximately one-half minute, that is, twice as fast as real time as with the analog computer and probably with considerably more accuracy.

Although indicative of things to come, these facts do not necessarily carry with them the implicit forecast that the digital computer will make the analog computer completely obsolete early in the next decade. For one thing, the cost of these ultra high-speed digital computers will be prohibitive for many uses.

It is likewise interesting to contemplate the future of analog-digital systems. Before the ultra high-speed digital computers appear in considerable numbers, it appears that the only possibility for the speed required for comprehensive real-time simulation of complex guided missile systems will come from these analog-digital combinations. It is entirely possible, also, that these systems will be very much competitive economically with the large-scale digital systems for many, many years to come. The analog-digital system carries with it the promise of flexibility for, obviously, the analog and digital computers can each be used separately on a wide variety of problems.

Many other possibilities of use appear. As examples, the digital computer can provide precision multiplication, function generating, or precision time-delay functions on a time-shared basis while it is performing computations independent of the analog computer. The availability of the conversion equipment will, of course, allow for playing FM-FM telemeter tapes into the digital computer for computations required in data reduction.

I would like to pose this question which I hope will be provocative: is it possible that the $200,000 analog-digital installation consisting of $50,000 of analog equipment, $50,000 of conversion equipment, and $100,000 digital computer will have more over-all computing potential than a $200,000 analog installation or a $200,000 digital installation? When one regards the digital computer as an equivalent number of precision multipliers, function generators, or precision time delays, the prospect may not seem too exciting; however, when one considers that the digital computer can be used for *all* of these things at different times as well as supply certain logical operations not possible on the analog computer, prospects seem considerably brighter.

It is platitudinous to remark on the importance of simulation to the complex electronic system to a technical audience. The expenditure of large sums of money for computing equipment to investigate guided missile characteristics is trifling compared with the money necessary to perform the equivalent number of firings of

[2] J. L. Greenstein, "Application of AD-DA Verter System in Combined Analog-Digital Computer Operation," presented at Pacific General Meeting AIEE, June, 1956.

[3] L. R. Turner and J. H. Rawlings, "Realization of randomly timed computer input and output by means of an interrupt feature," this issue, p. 141.

the actual missiles. I believe it can be stated unequivocally that the modern guided missile would not be possible without the modern simulation device. The actions of the complex system are only clearly and reliably understood when the complete mathematical formulation of the entire missile system has been accomplished and when the resulting equations have been solved a large number of times to determine the correct parameters and the correct detailed methods of missile guidance. It is tempting to say that the peaceful future of the free world depends on modern weapon technology and, therefore, depends directly on the electronic computer. As one with a heavy bias toward computers, I shall resist that temptation.

Digital Information Processing for Machine-Tool Control*

ALFRED K. SUSSKIND†

Summary—**Flexibility, accuracy, elimination of manual skill, and greater work potential result from the addition of digital data processing to machine-tool control. Several special-purpose digital computers for machine-tool control have now been developed. These perform only simple computational tasks. The more sophisticated tasks require the addition of general-purpose large-scale computers. This combination, while only partially explored, appears very promising. It should lead not only to improvements in translating a part design into the finished piece, but should ultimately assist in the design procedure.**

Introduction

FOR several decades now automatic machine tools in which the tool is controlled by analog methods have been successfully used. Here the instructions take the form of a master which is to be duplicated, or templates, cams, limit stops, etc. In the use of machines so controlled, considerable effort, both in labor and time, is incurred in tooling and setup. The first term refers to preparing the instructions (*e.g.*, making the masters) and the second refers to inserting them into the machine (*e.g.*, properly aligning several templates). Where production in substantial quantities is involved, the difficulties inherent in the analog control techniques have been of no serious consequence and for quantities such as are involved in automobile manufacture, special-purpose analog-controlled machines have amply demonstrated their high merit. But where production on a modest scale is involved, such as in the aircraft industry where perhaps a few tens of units are produced before design changes occur, there has arisen a need for new control techniques which are more flexible and more adaptable to rapid change. At the same time, demands for accuracy have become increasingly stringent. These needs can now be met through the use of techniques which control machine tools on the basis of digital information.

Adding digital information processing to machine-tool control results in several desirable features. First, the control instructions are in a form such as punched paper tape, cards, etc., all of which can be quickly changed when a different part is to be manufactured. Second, there is no limit to the accuracy with which the instructions can be specified or processed, and the limitations on the accuracy of the machined part are due entirely to the characteristics of the basic machine tool. Third, there is no manual skill involved in preparing the instructions such as there is in making masters, templates, etc. Fourth, there is no longer any limit on the machine-tool functions (number and type of tool motions, number of cutting tools, number and type of supervisory functions such as coolant ON-OFF, etc.) which can be brought under automatic control. Fifth, the machining can be performed at all times under desired optimum conditions (*e.g.*, any prescribed feed rate within the capability of the machine tool may be obtained). Sixth, the potential exists for including in the information processing more than the faithful translation of previously determined dimensions. While we do not know yet how to exploit this potential, it seems reasonable to expect that means might be found for letting the digital information-processing equipment contribute to certain parts of the design process. Thus, it would not appear out of the question to expect that, given the dimensions of mating surfaces and mounting requirements, the computer could design the part for minimum weight in light of prescribed stress requirements, as well as determine appropriate machine-tool instructions.

While this last point may not deserve the prominence given to it here, it serves to emphasize that the technology of digital machine-tool control is still far from

* Presented at Symposium on Computers in Simulation, Data Reduction and Control, New York, N. Y., March 21, 1957.
† Elec. Eng. Dept., Mass. Inst. Tech., Cambridge, Mass.

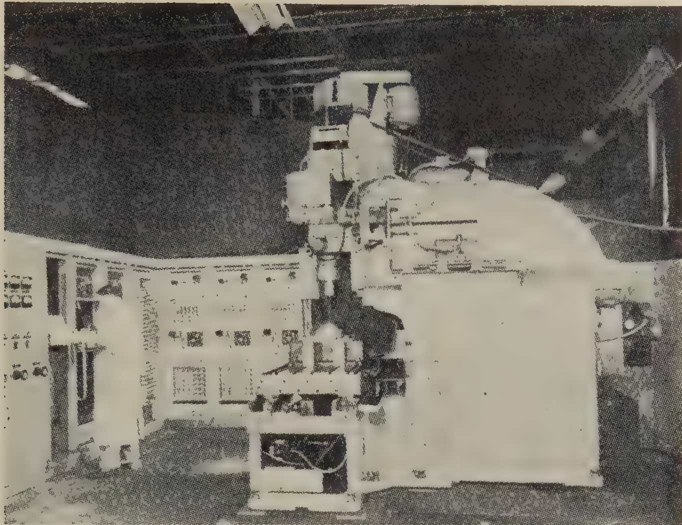


Fig. 1—Experimental numerical control system.

Fig. 2—Parts machined on experimental system.

completed. This is certainly not surprising, for the inclusion of digital information-processing equipment in a machine-tool system is a recent development. It started in 1949, when the Servomechanisms Laboratory of the Massachusetts Institute of Technology undertook the development of a numerically controlled milling machine.

The M.I.T. Experimental System

An experimental system was completed in 1952 and is shown in the photograph of Fig. 1. The machine tool is a standard model with three motions, x, y, and z, which form a Cartesian coordinate system. The information-processing system contained in the vertical panels accepts at its input a series of instructions, each of which contains four pieces of data, punched on paper tape: the dimensional increments Δx_j, Δy_j, and Δz_j through which the center of the cutting tool is to move from where the previous instruction has brought it, and the time interval Δt_j during which that motion is to be completed. The computing circuits then translate the instructions into a set of three pulse trains (one for each of the three motions), with as many pulses in each train as there are half-thousandths of an inch (the sensitivity

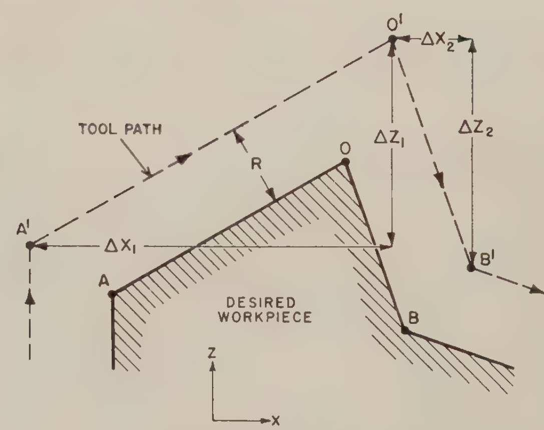


Fig. 3—Workpiece and corresponding tool path.

arbitrarily chosen here) in the corresponding increment, and with all three pulse trains nearly uniformly spread in time over the prescribed time interval. Finally, each pulse train is converted into an analog signal (here a shaft rotation), suitable as input to the corresponding drive mechanism on the machine tool. Since each instruction results in pulse trains and hence in shaft rotations and machine motions at a nearly constant rate, it follows that for each instruction the tool moves in a straight line. As a result, curves in this system are approximated by straight lines (*e.g.*, circles by polygons). Each straight line may be arbitrarily small so that the approximation can be made as good as desired.

A large variety of parts, most of them designed for aircraft, have been made on this experimental system. Some of these parts are shown in the photograph of Fig. 2.

It should be emphasized that because the controlled quantities are the machine-tool drives, the proper motions of the machine slides which will result in the desired cuts must be prescribed rather than the cuts themselves. Because of the nonzero dimensions of a milling tool, the slide motions are not equal to the desired cuts. This is illustrated in Fig. 3 for a planar series of cuts, where R is the radius of the cutting tool. To cut the path AB, the center of the tool must follow the path $A'B'$, and the specification of the latter is prescribed on the punched paper tape in the form of its components. Two sets of specifications are required for this example. If it is assumed that the path $A'B'$ lies in the plane of the machine-tool table, the instructions consist of Δx_1, Δz_1, and Δt_1 (the time during which the first motion is to be carried out), and then Δx_2, Δz_2, and Δt_2.

Since the machine-tool motions are the controlled quantities, the control system can assure only that the slide motions are carried out as specified. But this does not guarantee that the finished workpiece will have the desired dimensions; indeed, because of tool deflection, part warpage, etc., the workpiece will differ somewhat from that which would result from ideally rigid components. These deviations could be overcome only by monitoring the workpiece itself. However, conditions at

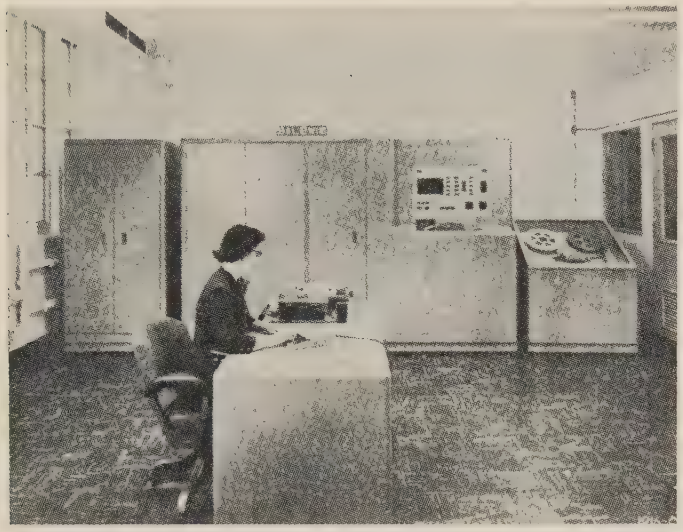


Fig. 4—Director system.

the cutting surface due to the presence of chips and coolant make feedback from the workpiece impossible. Not only does this restriction limit the accuracy with which parts can be made, but it also introduces an additional step in the information processing, for it requires that the dimensions of the desired part be translated into the proper tool path. Tool-center correction, as this translation is called, is thus an inherent requirement in information processing for milling-machine control by digital techniques.

In the experimental system, the information-processing equipment performs two functions: it carries out a linear interpolation between given points on the tool-center path in digital form, and it translates the result into an analog signal. There is no feedback between the machine tool and the director, as the information-processing equipment is called. It follows that separation of the director from the machine tool does not affect operation. If the output of the director were recorded on some medium such as magnetic tape and the magnetic tape were used to control the machine tool, the director would be free to prepare magnetic tapes for other parts or other machine tools, and it could operate in a convenient time scale with the magnetic tape playback speed appropriately adjusted to correspond to the difference in time scale between director and machine tool. This is the approach taken in a second director developed by the M.I.T. Servomechanisms Laboratory for the Giddings and Lewis Machine Tool Company and completed in 1955. Fig. 4 shows a picture of the equipment.

The Giddings and Lewis Director

While basically similar to the earlier experimental system, this second director contains several advances. A block diagram of this machine is shown in Fig. 5. The output of the director is a multichannel tape, with a group of six tracks containing the analog signals for controlling up to five machine-tool motions and the remaining seven tracks containing supervisory signals

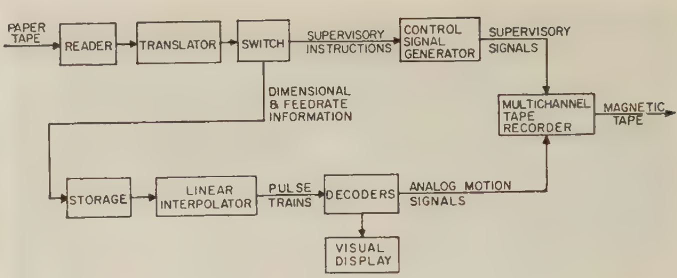


Fig. 5—Director system block diagram.

such as coolant ON-OFF, etc. The motion information is in analog form, expressed as phase shift of a 200-cycle carrier. Parameters are chosen so that for each 0.1000 inch of travel of a given machine-tool motion the corresponding signal is shifted 360 degrees in phase with respect to a reference 200-cycle signal which occupies the sixth track of the first group.

The reference and phase-shifted signals are generated electronically by a group of five decoders which receive at their inputs the corresponding pulse trains. In this machine too, each pulse train contains as many pulses as there are half-thousandths of an inch of travel in the desired machine motion and the pulse train is also nearly uniformly spread over the specified time interval.

Next in the block diagram is the linear interpolator. Its function is to generate the five pulse trains in response to the specified Δx_j, Δy_j, Δz_j, $\Delta\phi_j$, $\Delta\theta_j$, each spread over the same time interval Δt_j. The interpolator here is similar in principle to that used in the experimental director and is described by McDonough and Susskind [1].

The input to the director takes the form of punched-paper tape. Here the standard Flexowriter code, with a minor modification, is used, so that every digit of the instructions occupies one line of tape. Reading is accomplished by means of a standard Flexowriter reader, which makes information available serially, line by line. But the interpolator requires the entire instruction in parallel and, in order to avoid a discontinuity in the information processing, a new instruction must be immediately available in complete form when the previous set of pulse trains is completed. It is the function of the storage unit to retain an instruction until no longer needed and to assemble the next instruction by performing a serial-to-parallel conversion while the previous instruction is being interpolated. In the Giddings and Lewis director the storage unit is made up of square-loop magnetic material cores, assembled as stepping registers for the serial-to-parallel conversion.

The block diagram of Fig. 5 also shows a box labelled Translator. This unit carries out a translation between Flexowriter code and the internal language of the director. While both are in binary coded decimal form, the use of a different internal code (5-2-1-1) leads to greater computational ease inside the director. The unit marked Switch separates instructions designating the motions (Δx_j, Δy_j, Δz_j, $\Delta\phi_j$, $\Delta\psi_j$, Δt_j) from supervisory

instructions. The latter are directly translated by the Control-Signal Generator into tape-recording excitation.

Finally, there is a unit labelled Visual Display, which monitors the operation of the decoders. It puts on display, in decimal form, the distance in inches through which each of the five machine-tool motions has been commanded to move from the start of the program. This feature allows human observation of the information processing and is an aid in program debugging and director trouble shooting. Decimal representation was chosen here and in the input (paper tape) for convenient man-machine communication.

The director also contains a group of control circuits which perform such functions as power-supply control, initial clearing of the system, time-scale settings, and automatic slow-down of machine-tool commands when successive instructions would otherwise result in excessive accelerations. Means for manually modifying all programmed feed rates and inverting sign conventions are also provided. The latter permits processing of right- and left-handed parts from the same punched-paper tape.

Other Directors

Both of the directors described above perform but one major computational task: linear interpolation between points in space through which the center of the cutting tool moves. The question may well arise whether or not so limited an ability is adequate for milling surfaces. The answer is affirmative if the execution of useful work at reasonable cost is accepted as the criterion. But this is not to say that directors with higher computational ability might not be more successful.

For example, it is not difficult to imagine a director system which accepts at its input a series of points on the workpiece itself rather than on the required tool path, and which computes the required tool path that will result in connecting these points by a second degree curve such as a parabola or circle. This approach is described in the literature [2]. One could extend the computational ability of the director even further, so that the required input instructions are reduced to the point where either the workpiece is described only in terms of the equations of the surfaces which define it or, in the absence of such equations, by a series of key points such as are now given at successive stations of an airplane wing. In the latter case, the director would derive appropriate equations of any degree connecting them. This director should also be capable of choosing proper feed rates in view of machining requirements such as available spindle-drive horsepower and permissible cutting tool loads, and in view of required work tolerances and dynamic characteristics of the machine-tool drive servomechanisms. Clearly, an information-processing problem of considerable magnitude results. It appears that its solution requires the use of machines comparable to the large-scale computers (1103, 704, etc.) and that the flexibility of these machines is needed. One might then conclude that because these computers are already in existence and becoming more easily available there is no need for developing sophisticated machines aimed specifically toward the control of machine tools. It may well be that the standard computers, coupled with simple directors which perform the final data expansion (linear interpolation in the examples described above), form the optimum information-processing system for machine-tool control. The director is then akin to a computer terminal device which takes over the time-consuming, but fairly simple, last steps of the processing.

Directors as Computer Terminal Devices

Studies have started at the M.I.T. Servomechanisms Laboratory which are directed toward making use of large-scale computers in the information processing for machine-tool control. The first phase of this work, carried out by Siegel [3], has considered only the case of planar configurations consisting of circles and straight lines. It has resulted in the preparation of programs for the Whirlwind computer which enable persons with no previous knowledge of digital computers to use Whirlwind to prepare complete instructions for the experimental director. Given the specifications of the circles and straight lines which make up the desired part, the size of the cutting tool, the tolerance required, and the desired machine-tool feed rates, all stated in language closely approximating ordinary English, the computer has been programmed to prepare the control tape for the director. In its operation, the computer first translates the input language, chosen from the point of view of human convenience, into its own language and deduces the nature of the computations needed. Next, the computer determines the complete contour of the part to be made. In the third step, the computer finds the straight lines which approximate the contour within the accuracy specified. Then Whirlwind computes the cutter-center path corresponding to the straight lines. Next, the tool-center path, together with feed-rate specifications, is coded into the form required by the director. Finally, the complete commands are printed out on punched paper tape, ready for insertion into the director.

At the present time, the Laboratory is extending this work to the general three-dimensional case, without restriction on the types of functions which specify the desired surfaces. Computer programs will ultimately be written for generally available machines, *e.g.*, 704, rather than the Whirlwind computer. Another extension will consist of including the capacity to control machine tools with more than three motions. Again, the aim is to construct programs in such a way that they can be used by personnel with no previous computer experience. It is expected that these programs will be of maximum benefit if used by design or production personnel without familiarity with computer technology.

Studies in the area of utilization of general purpose

digital computers for determining director instructions are now being carried on at several places, all primarily concerned with aircraft production problems. This work is in the early stages, and it will be some time before significant results can be expected.

Other Types of Machine Tools

So far, the discussion has centered on milling machines. For these tools, information processing requirements are greatest. The path of the tool must be controlled at nearly all times since the tool is in contact with the work and is removing metal almost continuously. There is a large group of other machine tools in which the tool is in contact with the work only intermittently.

Consider, for example, a drill press making a series of holes in a plate. The tool is first placed over the desired location of the hole. Next, the tool travels into the work to the desired depth and is then withdrawn. In the next step, the tool is put into position for a second drilling operation, and the process continues. Note that there is no longer a requirement for path control, for after the tool withdraws from one hole and until it is ready to plunge in for the next hole, no metal is being cut. Thus the control system need only assure that before plunging starts, the tool is at the desired coordinate and the path taken after the previous withdrawal is of no consequence except that it affects the time spent between successive drilling operations.

It follows that for this drill press one need only specify the coordinates of successive holes, the depth of each hole, and the rate at which drilling is to occur. It is quite simple to design a control system which accepts these specifications in digital form, and even though the data processing required here is negligible, a digital control system may be advantageous because of the ease with which the input specification can be prepared, edited, and automatically and accurately read. Examples of such systems are given by other authors [4]–[6]. In these systems, the position instructions are read and then compared, either in digital form or in analog form after they have been decoded, with the feedback signal indicating the position of the controlled motion. The result of the comparison is used to bring the controlled motion into alignment with the specified position.

Future Developments

So far, all applications of digital information processing to machine-tool control have been concerned with the production of parts which were designed for conventional production techniques. As the new technology becomes more fully exploited, it will lead to modifications in established design procedures. For example, much effort is now being expended on producing drawings which are graphical representations of some of the part specifications. However, for digital information processing, a symbolic representation is needed, such as equations or tables of points defining the required part.

It seems reasonable to expect that the benefits of digital information processing can be extended beyond the area now explored which covers only a portion of the over-all process of translating the need for a part into the finished product. A more comprehensive approach should become the subject of future research.

Acknowledgment

The research leading to the experimental director and the studies in the use of digital computers for automatic programming of directors were carried out under sponsorship of the U. S. Air Force, Air Materiel Command.

Bibliography

[1] McDonough, J. O., Susskind, A. K., "A Numerically Controlled Milling Machine," in "Review of Input and Output Equipment Used in Computing Systems," *Proceedings of the Eastern Computer Conference*, Publication S-53, pp. 133–136.

[2] "A Three-Dimensional Machine Tool Control System," *Electronic Engineering*, Vol. 28 (May, 1956), pp. 204–207.

[3] Siegel, Arnold, "Automatic Programming of Numerically-Controlled Machine Tools," *Control Engineering*, Vol. 3 (October, 1956), pp. 65–70.

[4] "A Horizontal Boring Machine with Automatic Coordinate Setting," *Electronic Engineering*, Vol. 28 (May, 1956), pp. 193–195.

[5] Jaeger, J. J. "Numerical Control for Jig Boring Holds Production Promise," *The Tool Engineer*, Vol. 37 (July, 1956), pp. 102–107.

[6] Kelling, L. V. C. "Numerical Control of Machine Tools," IRE Trans. on Industrial Electronics, No. PGIE-2 (March, 1955), pp. 3–8.

[7] Susskind, A. K. "Bibliography on Numerical Control of Machine Tools," Servomechanisms Laboratory, Massachusetts Institute of Technology, Cambridge, Mass., February, 1957.

Realization of Randomly Timed Computer Input and Output by Means of an Interrupt Feature*

L. R. TURNER† AND J. H. RAWLINGS†

Summary—The Lewis Flight Propulsion Laboratory of the National Advisory Committee for Aeronautics is assigned the primary task of research into the basic problems of the design and operation of aircraft power plants. Since World War II this has primarily meant research in the field of jet and turbine engines.

Although these engine types are simple in concept, it was quickly realized that they could not be developed adequately by research on small-scale models. It has been necessary to build large facilities in which research could be conducted under conditions approximating those expected of actual flight.

With new facilities under construction or being planned, it became clear in the late 1940's that only some form of mechanization could cope with the growing problem of data-processing. The resulting studies led to a plan in early 1950 for the construction of a data-recording system aimed toward digital data-processing techniques.

In January, 1952, the first model of an automatic digital pressure recorder with mechanized, manually-controlled computing was put into operation.

From this point the system gradually evolved in concept and in working equipment to its present form in which a central editing and computing station serves a number of remote research facilities. The operation of data collecting and input editing is described in considerable detail by Ryskamp.[1] It is sufficient at this time to give a brief description of the system.

System Description

A CENTRAL Automatic Digital Data Encoder (CADDE) is connected by telephone cables to nine remote experimental stations. The CADDE system is designed to record up to 500 steady-state quantities in any record and, when it is available, automatically connects itself to the requesting facility and controls the sequence of operations required to record the desired voltage, pressure, and speed measurements. These are recorded together with identification and miscellaneous control information on magnetic tape, which then serves as a legal record of the experiment and also as an input store for data processing. The magnetic tape may also be used to produce high-speed plots of the unprocessed data or to type selected information on an electric typewriter.

The computing unit of the Central Data Processing System is a Remington Rand Univac Scientific Computer, Model 1103, which has been modified to adapt to the task of communicating with the data recorder and with remote devices for reporting of results, in order to achieve on-line data processing.

Mechanically, the most important problems to achieve on-line data processing arise from the wide difference between relatively high speeds required for editing and computing, and slow speeds of inexpensive devices for data transmission. Another factor is the awkward variation in the rate at which results may be produced. This rate may vary from perhaps one hundred per second to as low as one every three or four seconds.

Fortunately, the quantity of information which is to be displayed remotely for immediate inspection (performance coefficients or plots) is sufficiently small for adequate speeds to be obtained by multiplexing a few efficiently used slow-speed devices, such as the punches of Flexowriters, while the bulk output is directed toward higher speed devices for deferred printing.

This degree of multiplexing is, however, not the complete answer, because the sum of computing time and input-output time is still greater than desired. It is necessary to interlace computing and information transfers to achieve required speeds. It is well known that almost all of the cycle time of input-output devices is available for computing if coding which is clever enough can be prepared. In practice, the difficulty of such coding is so great that little use is usually made of this theoretically available time.

A satisfactory solution to these problems which requires no external buffers and generates no important coding difficulties was achieved by the development of the Lewis Laboratory's version of Interrupt, which was first put into operation in February, 1956.

In this version of Interrupt, programs in the high-speed store of the computer drive the input-output devices and provide for the necessary information transfers. These programs are in the form of subroutines which are referenced automatically at the time required by the device itself by means of the interrupt feature.

Interrupt is permitted only when the interrupting device is ready for an immediate information transfer. As each device for which interrupt has been enabled requires service, a signal to the computer causes the normal sequence of command procurement from the store to be interrupted and, at the same time, an arbitrary command is planted in the input register for immediate execution. The command counter is not advanced. In nearly all cases the injected command will be the Return Jump command of the 1103 with the addresses of the exit and entry lines of the appropriate service subroutine. A properly coded subroutine will provide

* Presented at Symposium on Computers in Simulation, Data Reduction, and Control, New York, N. Y., March 21, 1957.

† Lewis Flight Propulsion Lab., N.A.C.A., Cleveland, Ohio.

[1] J. H. Ryskamp, "System for multiple instrumentation outputs," *Automatic Control*, vol. 5, pp. 16–19; August, 1956.

for the next information transfer and will return computing to the point of interruption.

While experience has shown that the interrupt mode achieves high efficiency of all components without adding significantly to the difficulties of coding, there are, however, a few important ground rules which must be strictly followed, both in hardware design and in coding, if an interrupt system is to work at all. The importance of adherence to the standards was forcefully given point during the early operation in the interrupt mode. It was learned that a problem, or nearly all of a group of similar problems, could be solved correctly with incorrect service subroutines and with the actual omission of logically required hardware.

As an example of a coding error which might go undetected, it is clear that the failure to set an interlock against overwriting an output store before it has been translated and used would be unimportant if the output were always complete before the interlock was tested. Similarly, the complete failure of a hardware interlock intended to prevent a difficulty of rare occurrence might go undetected for a very long time.

The heart of these difficulties is a subtle difference between interrupt and noninterrupt modes of operation. In the conventional mode, not only does the result of calculation depend only on the initially stored code and on the data, but so does every intermediate state of the computer. With interrupt, because of the random timing of interruption, this is no longer true; in fact, even with a correct program, only the answers are expected to be predictable. One of the startling effects of this variability is that a programming or coding error, by producing a variable error or sometimes none at all, may be indistinguishable from a random computer malfunction. Many hours were spent in discovering this peculiarity.

The elimination of these hazards requires well-coordinated care both in the hardware design and in coding. The problem can be most easily stated from the coding standpoint by considering the successive tasks of an interrupt invoked subroutine. It is assumed that at the time of interrupt the computer is doing useful work.

The subroutine must first form a record of the content of the arithmetic registers (A and Q in the 1103) if they are to be used for housekeeping calculations. It must then prepare for the next step of information transfer and correctly modify any interlocks required to keep the input-output and the computational routines in step. Finally, it is necessary to restore the status of the arithmetic registers and the program counter. A further interrupt by the same device may not be permitted until these tasks are completed. If an earlier interrupt were permitted, the program reference stored by the return jump would be to some point in the service routine and an unintended tight loop would result.

From the hardware standpoint it is necessary to maintain an electronic connection between the input or output device and the computer until the transfer is complete. It is also necessary to prevent an interrupt by any device that would refer to the subroutine in current use. In the 1103 these conditions were easily met. The information transfer command is used to enable a further interrupt and, at the completion of its execution, cause the current input-output device to be disconnected. The information transfer command is the next-to-last command in the service routine. The last command is a jump which restores the program counter to its content at the time of interrupt.

The hardware contains a delay in enabling the next interrupt for one command after the information transfer command, thereby assuring that the problem status can be completely restored. The command structure of the 1103 is compatible with these simple controls because the arithmetic registers need not be used for input and output.

The electronic hardware executes several other important functions. With several interrupting devices, it is necessary to control through coding the ability of each device to initiate interrupt. This is done in the 1103 by means of the External Function (EF) command. One EF command is executed to set an enabling flip-flop when a task is generated. A different EF command located in the service routine resets this flip-flop when the task is finished.

Other functions of the hardware include the selection of a single device for interrupt when several may be competing for this service, and the usual safeguards against premature or improperly repeated information transfers.

At the present time the interrupt feature is used by the control and reading mechanism of the tape handlers on the CADDE recording system and for two Western Electric high-speed tape punches. Equipment is nearing completion to add four slow-speed punches, which will be used to transfer information through paper tape buffering to remote typewriters and digitally operated point plotters. Further details are given in the following sections.

Interrupt

When computation demands input or output, devices external to the computer are activated for interrupt. Any device, selected under program control, may interrupt the computer if it is ready for an immediate transfer of information. By interrupting the computer the normal sequence of command procurement is suspended and, instead, the interrupting device causes a command to be injected for immediate execution. Injected commands are emitted from a pluggable COMMAND EMITTER having a position available for each device.

The command selected for injection is a return jump command because its execution results in recording the contents of the Program Address Counter on the occurrence of interrupt. The injected commands provide entries to the service subroutines for the interrupting devices.

Devices which are ready for immediate information transfer compete for computer time. Conflicts arising from the fact that several devices may be ready at the same time are automatically resolved by an ANTICOINCIDENCE NET.

Operation

The system is illustrated in a block diagram [Fig. 1(a) and 1(b)], on the next page. Only two devices are shown because of space limitations.

Consider now that the computer is in the normal mode of operation, executing commands from consecutive address locations in storage. An EXTERNAL FUNCTION COMMAND is executed, turning on the ability of the OUTPUT DEVICE to interrupt. If there has been a previous use of this device, a GATED TRANSFER COMPLETE will have cancelled the READY SIGNAL generated in the previous cycle.

The OUTPUT DEVICE is not yet READY. When the OUTPUT DEVICE can accept new information immediately (its INPUT THYRATRON REGISTER cleared of previous information, and plate voltage reapplied), it sends a DEVICE RESUME signal to the READY CONTROLS. Having now received a signal turning INTERRUPT on, a GATED TRANSFER COMPLETE, and a DEVICE RESUME, the ready controls can generate a READY signal. If the ANTICOINCIDENCE NET is empty (no established priority), the READY signal is generated on the next NET ENTRY PROBE. If a second device has meanwhile interrupted the computer, the ANTICOINCIDENCE NET may not be empty. In this case the READY signal will not be generated until a GATED TRANSFER COMPLETE to the second device's READY CONTROL cancels the second device's READY signal and releases the ANTICOINCIDENCE NET.

NET ENTRY PROBES are not generated while there is an established priority in the ANTICOINCIDENCE NET. This insures that once a priority is established it is not disturbed until the device which has the priority is served with its share of computer time. After the transfer is complete, all devices which are READY compete for computer time.

The READY signal from the OUTPUT DEVICE READY CONTROLS is sent to the ANTICOINCIDENCE NET. If there are no other READY signals in the net which are higher than the READY signal from the OUTPUT DEVICE, then the OUTPUT DEVICE will receive PRIORITY.

The PRIORITY signal is sent to SEQUENCE CONTROL. An enable is generated in SEQUENCE CONTROL associated with the OUTPUT DEVICE and sent to the COMMAND EMITTER, which in turn enables the INPUT MIXER in preparation for the injection of a command into the INPUT-OUTPUT REGISTER. The COMMAND EMITTER is pluggable, and is plugged in the position reserved for the OUTPUT DEVICE with a return jump command. The v-address portion of the plugged command is the location in storage of the subroutine which will serve the OUTPUT DEVICE. The u-address portion is plugged with the address of the last command in the service subroutine.

An INTERRUPT request is generated and sent to MAIN CONTROL of the computer where it will request an alteration of the command procurement portion of the computer cycle. The request may be denied temporarily if the INPUT-OUTPUT REGISTER is in use or if the computer is in a repeat sequence. When the request is granted, the INTERRUPT signal transfers the contents of the INPUT MIXER (the injected return jump command) to the INPUT-OUTPUT REGISTER. The injected command is then sent from INPUT-OUTPUT REGISTER to ARITHMETIC EXCHANGE REGISTER and then into the main control where it is executed.

The INTERRUPT signal is also sent into SEQUENCE CONTROL where it initiates an INTERRUPT lockout. The lockout is not released until SEQUENCE CONTROL receives a TRANSFER COMPLETE (WRITE) from main control of the computer. This lockout prevents the OUTPUT DEVICE from interrupting its own service subroutine.

When the injected return jump command is executed, the computer records the address of the command which would normally have been the next to be fetched for execution in the interrupted problem. It then jumps to the address given by the v-address portion of the injected return jump command and finds the subroutine which services the OUTPUT DEVICE.

The next-to-last command in the subroutine is an information transfer instruction. A WRITE ENABLE is generated as a result of the execution of this transfer instruction and sent to SEQUENCE CONTROL. SEQUENCE CONTROL directs the WRITE ENABLE to enable the OUTPUT DEVICE THYRATRON REGISTER. The TRANSFER COMPLETE (WRITE) is also a by-product of the execution of the transfer instruction and is sent to SEQUENCE CONTROL. SEQUENCE CONTROL directs this signal to the READY CONTROLS of the OUTPUT DEVICE as a GATED TRANSFER COMPLETE, cancelling the READY signal. The timing of INTERRUPT and TRANSFER COMPLETE signals prevents a further interrupt before the execution of the last command in the service subroutine. This guarantees that the interrupted problem is always resumed if the last command of the sburoutine is a jump to the re-entry address of the interrupted problem.

Any device which is READY and HIGH in the ANTICOINCIDENCE NET may now interrupt the computer. Interrupt for the OUTPUT DEVICE is still turned on and it will interrupt again when it is READY. To turn off its ability to interrupt would require execu-

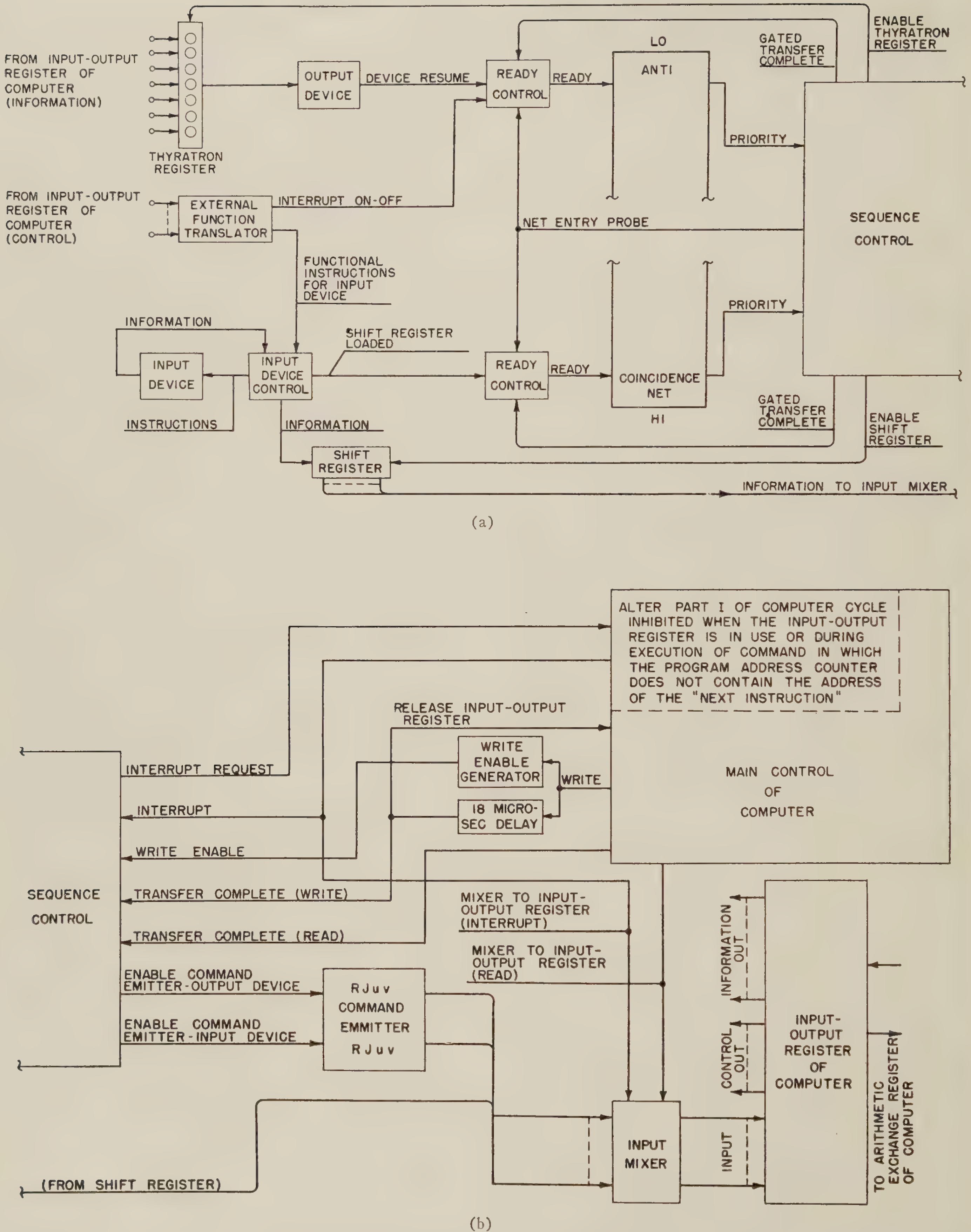


Fig. 1—Interrupt communication of INPUT and OUTPUT DEVICE with Remington Rand 1103 computer block diagram.

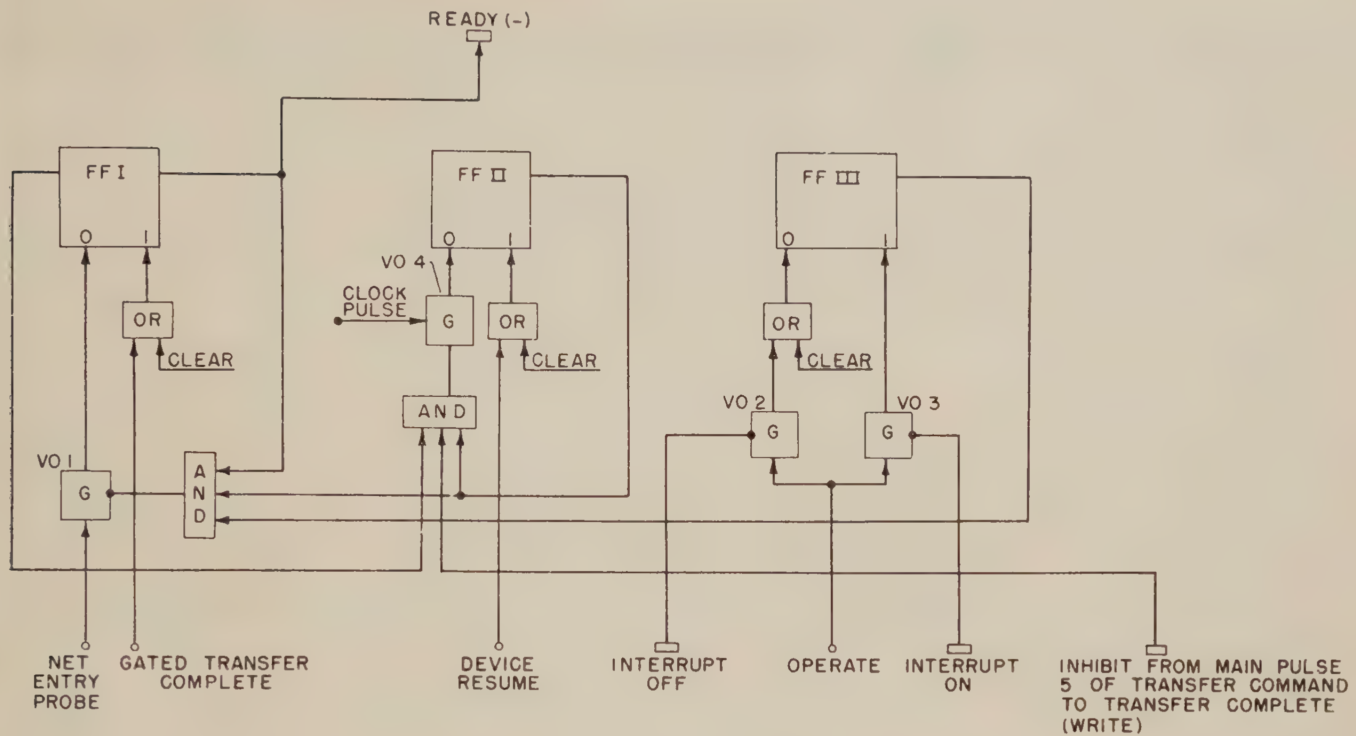


Fig. 2—READY CONTROL.

tion of another EXTERNAL FUNCTION command. This command is normally located in the subroutine servicing the OUTPUT DEVICE.

Ready Controls

READY CONTROLS are necessary to insure that information is transferred only when immediate acceptance is possible and that only one transfer is made per device cycle.

READY, for a particular device, means that it can accept or supply new information immediately. In operation all devices in the system are tested periodically to determine if they are ready. The testing is done by the NET ENTRY PROBE. The circuitry for this testing procedure is illustrated in Fig. 2.

When a device interrupts the computer and is serviced by the execution of a subroutine, the execution of the information transfer command within the subroutine results in a GATED TRANSFER COMPLETE to the READY CONTROLS of the interrupting device.

The GATED TRANSFER COMPLETE turns the READY signal off after its use so that the device cannot interrupt more than once per device cycle. When a device has completed its mechanical work, or when its buffer register is clear and ready for more information, a DEVICE RESUME is generated and sent to the READY CONTROLS of the device. Requiring a DEVICE RESUME insures that the READY signal cannot be turned on again until the device has accomplished its assigned task.

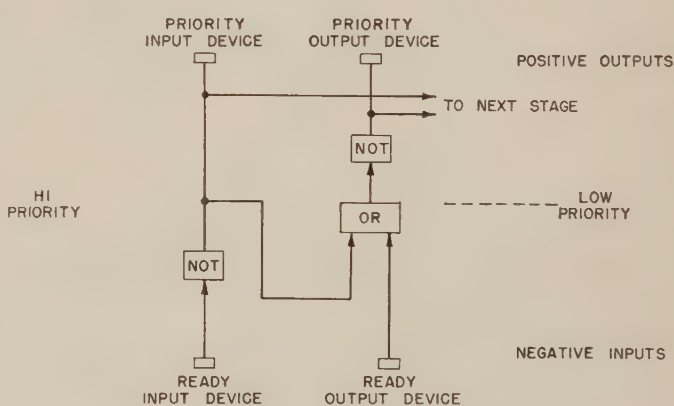


Fig. 3—ANTICOINCIDENCE NET.

Each device in the system must have its own READY CONTROLS

Anticoincidence Net

The ANTICOINCIDENCE NET is necessary because more than one device can be READY at the same time.

Each device in the system is assigned a position in the ANTICOINCIDENCE NET, which has priority assigning properties.

READY signals from device READY CONTROLS are sent to the ANTICOINCIDENCE NET illustrated in Fig. 3. An output from any position in the net cancels any other outputs from the net lower than itself. With multiple inputs, only the output which is highest in the net survives.

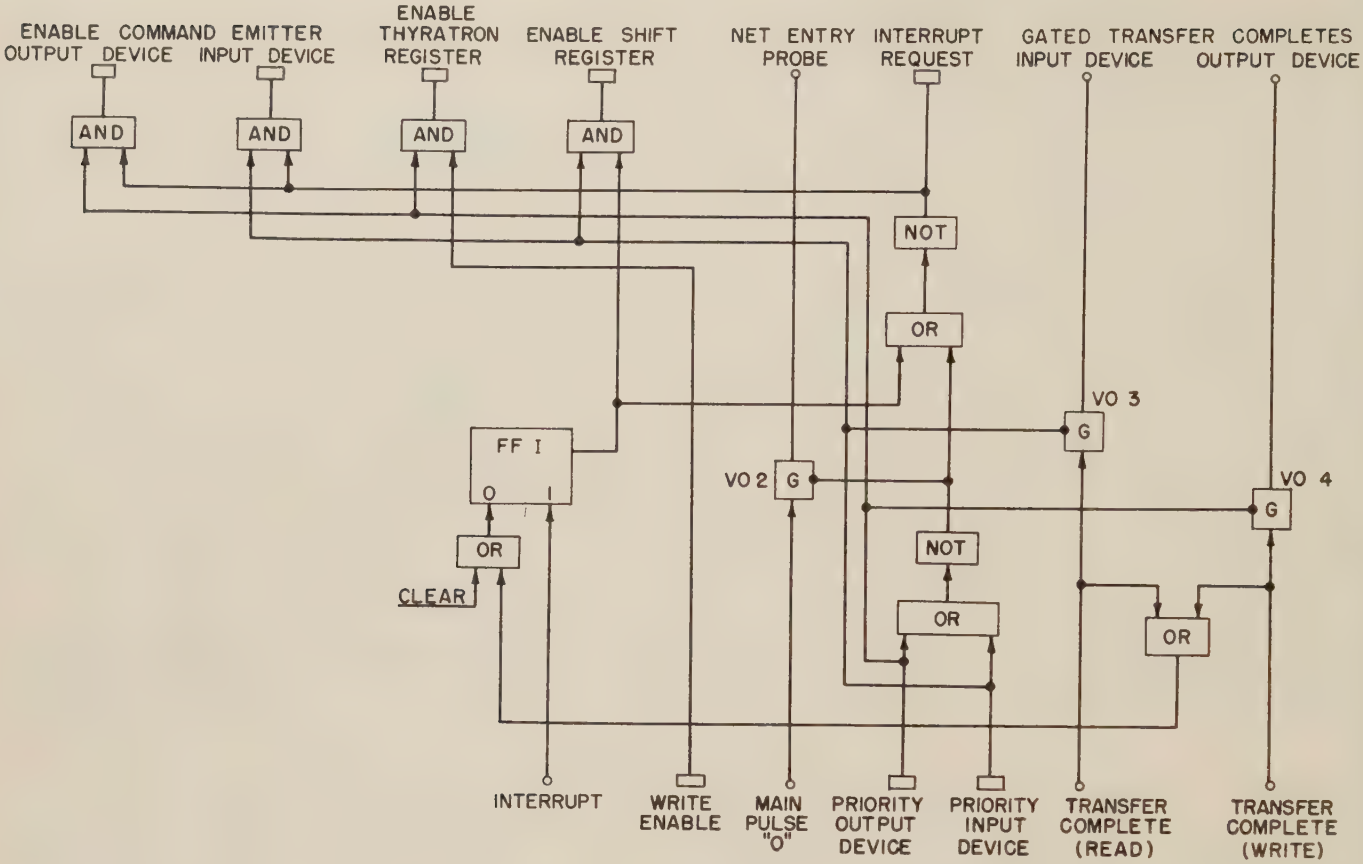


Fig. 4—SEQUENCE CONTROL.

Sequence Control

PRIORITY signals are sent to SEQUENCE CONTROL, which is essentially an information traffic director. It directs INFORMATION TRANSFER ENABLES and TRANSFER COMPLETE signals to the proper place and at the proper times.

For command injection, it directs a COMMAND EMITTER ENABLE to the pluggable position of the COMMAND EMITTER reserved for the interrupting device at the time appropriate for command injection. At the appropriate time for information transfer, SEQUENCE CONTROL directs a TRANSFER ENABLE to the input or output register of the device receiving or delivering information. When a transfer has been completed, SEQUENCE CONTROL directs a complete signal to the READY CONTROL of the device which has been served with computer time.

The circuitry is illustrated in Fig. 4. The flip-flop shown is initiated by the occurrence of INTERRUPT and reset by any TRANSFER COMPLETE. Its function is to sequence information transfer enables in the order in which they are to be used (for multiple transfers, a multistage flip-flop distributor would be required)

INTERRUPT REQUEST is generated in SEQUENCE CONTROL. It is locked out anytime FF I is set to "1," preventing any device from interrupting its own service subroutine.

NET ENTRY PROBES are generated in SEQUENCE CONTROL and sent to all READY CONTROLS. NET ENTRY PROBES are cut off by the presence of a priority signal.

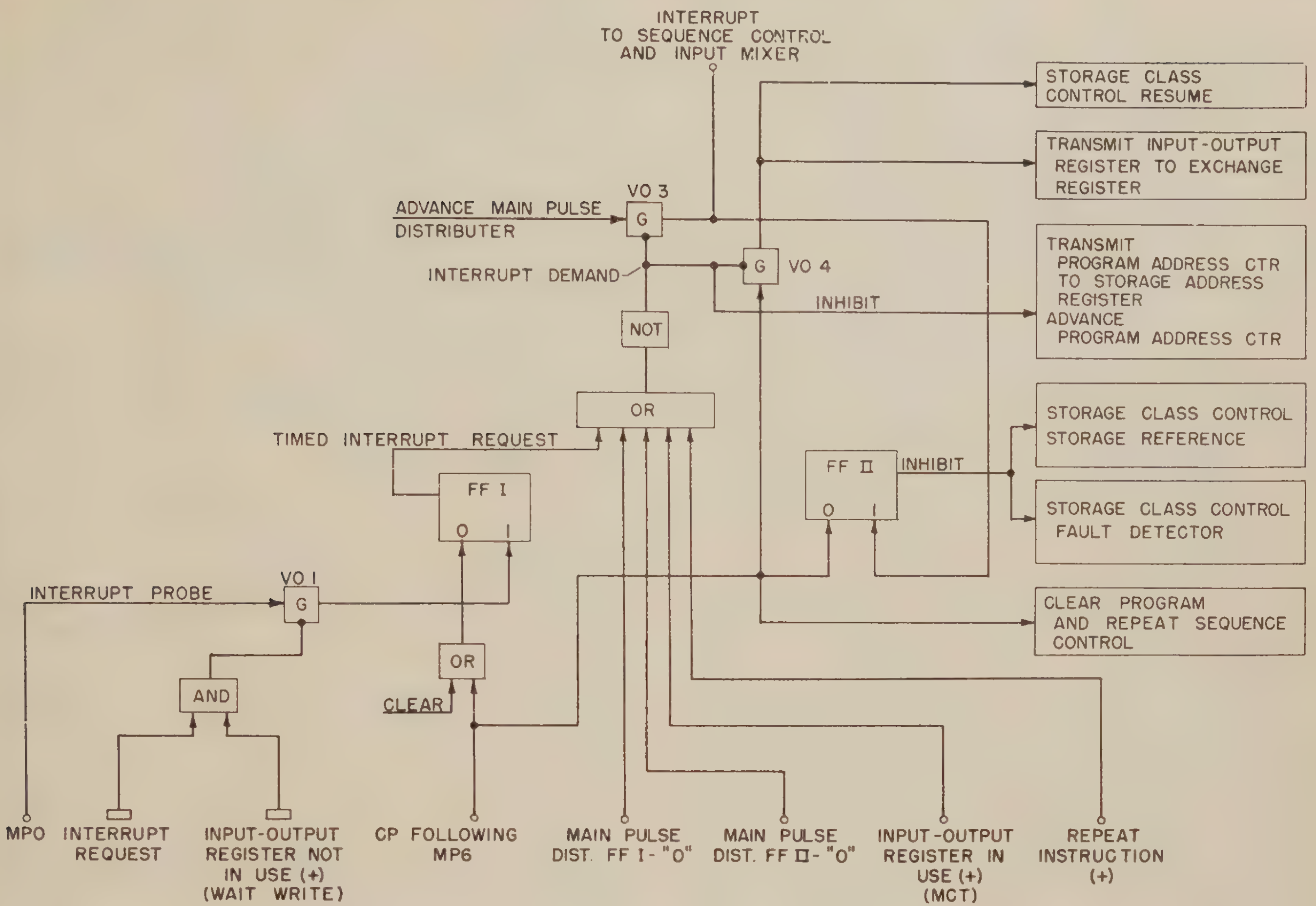
Expansion of SEQUENCE CONTROL to include more devices requires additional AND circuits to supply one COMMAND EMITTER ENABLE for each device and one INFORMATION TRANSFER ENABLE for each device. One TRANSFER COMPLETE GATE is required for each device, and the OR circuit receiving device priorities must have an entry for each device.

Alteration of Computer Cycle

Alteration of part 1 of the computer cycle results in fetching the next command to be executed from the INPUT-OUTPUT REGISTER rather than from the next consecutive address location in storage.

The normal functions of command procurement occur on MAIN PULSE 6 and MAIN PULSE 7. When INTERRUPT occurs, the functions of MAIN PULSE 6 are altered, but MAIN PULSE 7 retains its normal function.

Alteration of the computer cycle is illustrated in Fig. 5(a) and 5(b). If the INPUT-OUTPUT REGISTER contains information from a previous use or if a command is being executed which will make immediate use of the INPUT-OUTPUT REGISTER, then a request for INTERRUPT is denied.



(a)

	INTERRUPT		NORMAL
MP	Housekeeping	MP	Housekeeping
6	Transmit INPUT MIXER to INPUT-OUTPUT REGISTER Set FF I, Fig. 4=1 Set FF II, Fig. 5(a)=1 Initiate READ Set WAIT INTERNAL Clear X	6	Transmit PROGRAM ADDRESS COUNTER to STORAGE ADDRESS REGISTER Advance PROGRAM ADDRESS COUNTER Initiate READ Clear X WAIT INTERNAL
CP	Transmit INPUT-OUTPUT REGISTER to EXCHANGE REGISTER Clear FF I, Fig. 5(a) Clear FF II, Fig. 5(a) Clear PROGRAM CONTROL REGISTER Clear REPEAT SEQUENCE CONTROL STORAGE CLASS CONTROL resume	CP	Clear PROGRAM CONTROL REGISTER Clear REPEAT SEQUENCE CONTROL
		7	Clear STORAGE ADDRESS REGISTER Transmit EXCHANGE REGISTER TO PROGRAM CONTROL REGISTER Initiate 2-μsec delay
7	Same as NORMAL		

(b)

Fig. 5—Alter part 1 of computer cycle.

A request for INTERRUPT is denied when the computer is in a repeat sequence because the PROGRAM ADDRESS COUNTER does not contain the address of the next instruction to be fetched.

The CLEAR for the PROGRAM CONTROL REGISTER and REPEAT SEQUENCE CONTROL has been shifted unconditionally from MAIN PULSE 6 to the following clock pulse so that denials to INTERRUPT REQUEST, based on the contents of the PROGRAM CONTROL REGISTER or REPEAT SEQUENCE CONTROL, are effective throughout the time when INTERRUPT can occur. The shift in timing

does not affect operation of the computer in the NON-INTERRUPT mode. The purpose of FF I is to time the INTERRUPT REQUEST. In case the request is denied, it must be removed so that the normal sequence is not upset. FF I is reset unconditionally on the clock pulse following MAIN PULSE 6.

INTERRUPT REQUEST and INPUT-OUTPUT REGISTER IN USE (+) are ANDED to enable GATE VO 1, so that the timed request does not occur if there is a WAIT WRITE. This is necessary because of the random timing of devices that clear WAIT WRITE and are not included in the INTERRUPT system. The entries to the OR circuit from the MAIN PULSE DISTRIBUTOR FLIP-FLOPS are permissive and specify the time at which INTERRUPT can occur (MAIN PULSE 6 and the following clock pulse).

Generation of INTERRUPT DEMAND results in an INTERRUPT.

INTERRUPT DEMAND inhibits the normal housekeeping function of sending the contents of the PROGRAM ADDRESS COUNTER to the STORAGE ADDRESS REGISTER and advancing the COUNTER. The COUNTER is left intact on MAIN PULSE 6 when INTERRUPT occurs. On MAIN PULSE 6 (ADVANCE MAIN PULSE DISTRIBUTOR) the contents of the INPUT MIXER are sent to the INPUT-OUTPUT REGISTER, READ is initiated (but not allowed to go to completion), and FF II is set to 1. The purpose of FF II is to inhibit the reference to STORAGE initiated by the READ and to inhibit the detection of a storage class control fault. The fault would occur as a result of inhibiting the reference to storage, since the inhibition results in no storage class selection.

On the clock pulse following MAIN PULSE 6, the contents of the INPUT-OUTPUT REGISTER are sent to the ARITHMETIC EXCHANGE REGISTER (X), and a storage class control resume is generated (WAIT INTERNAL is set on INITIATE READ). Flip-flop II is reset.

On MAIN PULSE 7, the injected command is sent from the ARITHMETIC EXCHANGE REGISTER (X) to the PROGRAM CONTROL REGISTER for execution. This is a normal function for the 1103.

Ground Rules

The following rules characterize the system and also indicate some of the design considerations.

1) A device may not interrupt its own service subroutine. INTERRUPT REQUEST (SEQUENCE CONTROL) is locked out when INTERRUPT occurs and this lockout is not released until a TRANSFER COMPLETE is given. The TRANSFER COMPLETE is a by-product of executing a transfer command. Therefore, the location of the transfer command, within the service subroutine, is important. It must immediately precede the exit from the subroutine, unless it can be shown that the device in question cannot become READY again before the exit is made.

2) A device may not interrupt more than once per device cycle.

3) A device may not use time-shared equipment for longer than is necessary for acceptance of the interrupt request and execution of the service subroutine.

Cancellation of the READY signal by GATED TRANSFER COMPLETE and release of the INTERRUPT lockout (SEQUENCE CONTROL) by TRANSFER COMPLETE insure rules 2 and 3 above.

4) A device may not interrupt if there is an incomplete previous assignment. This means completion of the actual mechanical assignment. DEVICE RESUME is the signal indicating completion and is required to produce a READY signal.

5) An established priority may not be disturbed until it is cancelled after use. NET ENTRY PROBES are not generated while there is an established priority.

6) If INTERRUPT is to be turned off for a particular device the EXTERNAL FUNCTION command for this purpose must be located within the service subroutine, unless it can be shown that the EXTERNAL FUNCTION command will be executed before the device can become READY again.

Timing and Interlocking

Time allowed for switching transients in the ANTICOINCIDENCE NET and SEQUENCE CONTROL is determined by the timing of the following pulses:

1) NET ENTRY PROBE—MAIN PULSE 0 and ANTICOINCIDENCE NET empty.
2) TRANSFER COMPLETE (READ)—MAIN PULSE 2 and EXTERNAL READ (MCT 76).
3) TRANSFER COMPLETE (WRITE)—first MAIN PULSE 0 to occur after write transfer instruction is executed.
4) INTERRUPT probe—MAIN PULSE 0.

The minimum time allowed is given in the following situation:

1) The ANTICOINCIDENCE NET has an entry (READY signal) from a device competing for computer time.
2) An input device with higher priority is interrupting the computer.

The TRANSFER COMPLETE which releases the ANTICOINCIDENCE NET is given on MAIN PULSE 2 of the READ transfer. The READY signal is waiting to establish a new priority. The next interrupt probe to occur will determine the earliest possible time that an attempt will be made to make use of the newly established priority. The next interrupt probe will occur on MAIN PULSE 0 of the command following the transfer instruction. The minimum time will be 22 μsec.

When the information transfer instruction is executed in a service subroutine, the timing must guarantee that

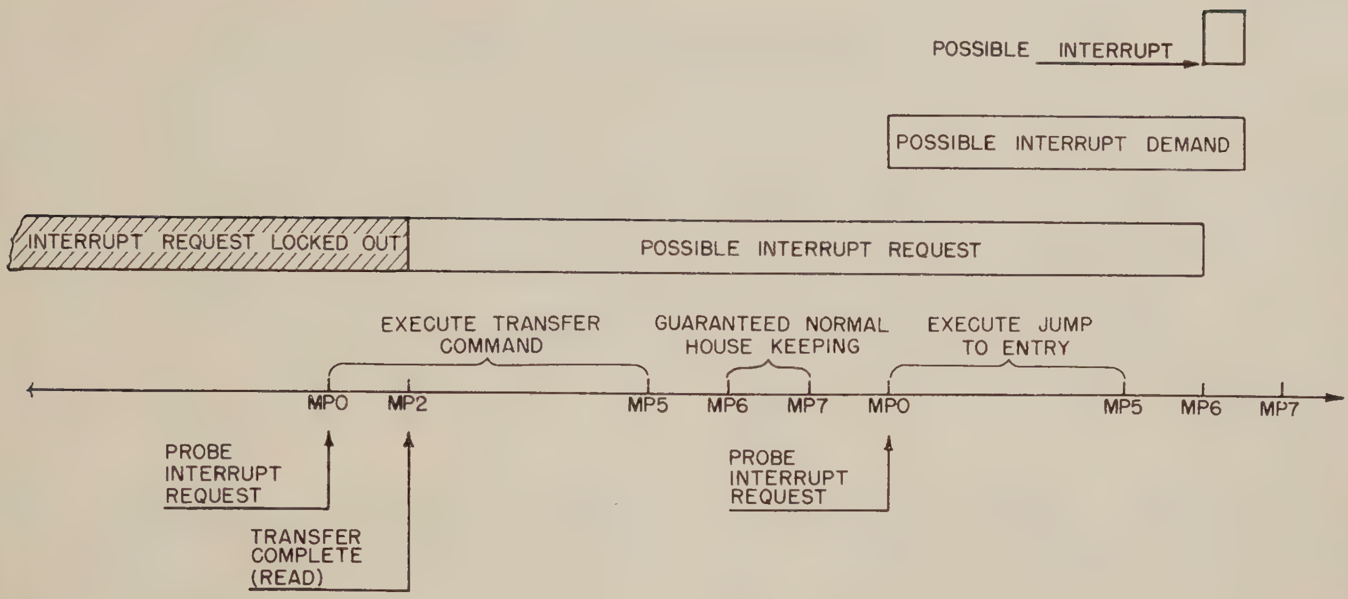


Fig. 6—Timing.

at least one more command can be executed before a further interrupt.

The SEQUENCE CONTROL lockout of INTERRUPT request is relased on MAIN PULSE 2 of the READ transfer command. The preceding MAIN PULSE 0 probed INTERRUPT REQUEST and found none because of the lockout. INTERRUPT REQUEST can be generated after MAIN PULSE 2 of the READ transfer instruction, but will not be probed to set up an INTERRUPT DEMAND until the following MAIN PULSE 0. Normal command procurement is therefore guaranteed on the MAIN PULSE 6 following execution of the READ transfer command. The timing is illustrated in Fig. 6.

Programming

Program interlocks must be set up to guard against several situations which will arise. For output, storage space is reserved to stack the results of computation. As soon as the results are stacked in this file, an output device is assigned to empty it. This is done under program control by executing an EXTERNAL FUNCTION command turning INTERRUPT on for the output device. When the file becomes empty, INTERRUPT must be turned off. Turning INTERRUPT on too soon or off too late will result in some nonsense in the output. An interlock must also be provided which prevents overwriting the file with new results before it has been emptied of previous results. If computation is finished before a file has been emptied of results, the computer must be kept running until the output job is complete. Provisions for saving the contents of the ARITHMETIC REGISTERS and remaking them before reentering the interrupted problem must be included in the service subroutines.

The nature of INTERRUPT with randomly-timed devices is such that the thin line which has separated program difficulties from hardware difficulties is now invisible. A problem with a program error may be successfully run many times before the peculiar circumstances which produce a failure show up. When they do show up there is little or no evidence to be analyzed.

Conclusion

Communication of external input and output devices with the Remington Rand 1103 computer has been accomplished on an INTERRUPT basis. The devices compete for computer time whenever they are ready. Service subroutines stored in the memory of the computer are referenced by the devices themselves when INTERRUPT occurs.

A device may not solicit computer time unless it is ready for an immediate transfer of information. The computer is never required to stop and wait for some mechanical work to be accomplished. The devices in the system are required to do all preparatory work "on their own time."

Development of a system of communications based on INTERRUPT, with program controlled use of internal storage as a buffer, has eliminated the undesirable effects of entirely sequential programming of input, computing, and output. Multiple device INTERRUPT permits a mixture of input, output, and computing, at optimum speeds for all.

The experience thus far with interrupt has been very favorable. In the first use of the feature, which was on an engineering problem with a large output, effective computer speed was very nearly doubled. A somewhat better improvement is expected in data processing problems when the full ration of equipment is installed.

Questions and Discussion*

The Symposium on Computers in Simulation, Data Reduction, and Control held at the 1957 National Convention of the Institute of Radio Engineers, was sponsored by the Professional Group on Electronic Computers and presented in the Starlight Roof of the Waldorf-Astoria on March 21, 1957. Five invited papers were followed by questions and discussion between the audience and a panel that included the authors, except E. L. Braun who could not be present, Harold K. Skramstad of the National Bureau of Standards, Morris Rubinoff of the University of Pennsylvania, Eric Wolf of Lincoln Laboratory, Walter F. Wakefield of the General Electric Company, and the session organizer and chairman, R. D. Elbourn of the National Bureau of Standards.

Chairman Elbourn: In choosing the subject of this symposium we took our cue from a remark of Jay Forrester, who pointed out that between 1945 and 1950 we were learning how to build the first computers; from 1950 to 1955 we were primarily concerned with solving mathematical and engineering problems; and in the period that we are in now—1955 to 1960—the principal interest is in business applications and data processing. He predicted that from 1960 to 1965 our principal interest would be in automatic control applications of computing machinery.

We had to use a long and awkward title for this session because there did not seem to be any one word which described this group of applications, although they have very recognizable characteristics in common. Most notable, perhaps, is the fact that the computer is no longer the prima donna at whose beck and call everything else starts and stops. Instead, we now have the computer dancing to the tune called by some other part of the system. It may have to be running in real time; it may be interrupted; it may have to respond to an unexpected need of the process control in a plant. Usually we find both analog and digital information in the same system, and thus the necessity of converting from analog to digital and from digital to analog; and often we find the need for communication between men and machines.

Question from the Audience: On what experience do you base the statement of one failure per hour for present-day digital computers?

Chairman Elbourn: If I recall, the figure was one failure per day, and an average of about 23 hours and some minutes has been achieved by the Whirlwind. Perhaps the author had some others in mind.

Mr. Gunning: Whirlwind is probably the best example. I tried to get some examples from people who are using commercial machines, and I got the general impression that a few hours is more appropriate for the regularly available machines of the large class. I am not talking about the smaller machines.

Question: What do you call a small machine?

Gunning: I will say what I mean by a large machine and exclude all the others. I mean machines of the IBM 700 series or the Whirlwind—machines of that type.

Dr. Rubinoff: I think the comment made by one of the speakers, Mr. Bauer, was an interesting one. He raises what I consider an important question, namely, the optimum interconnection of economical units of different types of computers, where I guess by "economical units" I mean the small computers; I will exclude the large ones. In particular, I was thinking about one speaker, who favors the use of a DDA and perhaps even the use of a digital computer and DDA interconnection. Another speaker favors the digital and the analog interconnected through a converter.

The obvious question is, what would we win if we tied together a digital computer (a relatively inexpensive one), a DDA, and an analog computer, *i.e.*, all three relatively inexpensive computers? Would there be any real advantage from the standpoint of the amount of work per dollar? Would there be any inherent disadvantages?

Dr. Bauer: With the present state of the digital differential analyzer, I don't feel that it is a good competitor with the more conventional analog computer in an analog-digital hybrid system. However, I understand that at least one manufacturer is coming out with a digital differential analyzer that has a much higher integrating frequency, by an order of magnitude, as I recall. In this case I think the prospect of using conventional general-purpose electronic computer techniques with differential computer analyzer techniques seems very excellent.

I don't know why anyone would ever want to put the conventional analog computer in with the DDA and the GP computers, provided that, as I said, the digital differential analyzer were of the more modern, advanced type which, unfortunately, is not on the market right now.

Mr. Braun:[1] I would like to return to a point mentioned in my paper. It was stated there that absolute

* Manuscript received by the PGEC, February 14, 1958.

[1] Since Mr. Braun was not present, his contributions to the discussion were added during editing of the transcript.

computation was appropriate in some areas and incremental computation was better in others. There are many ways of combining absolute and incremental computing elements into a computing system. As yet, no proof has been offered of what is generally the best combination, or even that there is a "best" combination. The problem hinges on two important considerations: the division between relatively high- and low-speed computing requirements in a system, and the components available for mechanizing the computer. Several absolute-incremental hybrid types of computers have been proposed and/or are being built. However, not enough experience has been gained to allow one to state with certainty the nature of the "best" computer even for a given class of applications. If one could determine this for elements currently available, the answer would be subject to change as new elements appeared. The following situation could occur. The appearance of certain new elements might dictate that one abandon a logical design of type "A" in favor of one of type "B." Later, the appearance of other elements could cause one to go to a design of type "C" or possibly to revert to a design of type "A."

It may be expected that such oscillations in design philosophy may continue for some time to come.

I agree with Dr. Bauer's questioning of the advisability of combining a DDA and a GP digital computer with an analog computer. In all probability, DDA's can be developed that can cause the obsolescence of present analog computers for such applications as the simulation of high-speed aircraft and missile performance.

Rubinoff: There was a question in the middle of Dr. Bauer's answer, and I would like to say one word on that. He said that he didn't know why an analog computer might want to be tied to a digital computer if it has a DDA. In the design of a control system, you may have to include analog components such as servomechanisms, in which case you might just as well include some analog computation.

We ought to be a little clearer as to what we mean by an analog computer. It seems to me that a complete control system has to have such things as motors and sensory equipment. The sensory equipment might well be digital, although I suspect that it would be analog. I guess you can make digital motors, but I don't know of any effort to develop them at this time. I feel, therefore, that all three do tie in together.

I would also like to comment on the other speaker who mentioned the DDA. I got the impression from speaking to someone recently that there is a real advantage in including a DDA even in a digital computer of high speed. The advantage of the DDA is that once the necessary derivatives are available, the actual integration process of a slow DDA is comparable in speed with the integration time of a high-speed digital computer, because of the way the information is processed in the DDA to obtain optimum efficiency of the computing units.

Braun: Dr. Rubinoff has just raised two questions and has also presented speculations on the answers. With the first speculation I disagree. With the second I concur, and would like to add some substantiating comments.

First, it does not follow that because you happen to have analog sensors and effectors you should also have some analog computation. Analog computation would have to be justified by taking other criteria into consideration, *e.g.*, such things as reliability, cost, required speed of computation, precision, etc.

Second, it is true that there are advantages to adding a DDA to a GP computer. One of the points that I have tried to stress is that the DDA is a very efficient means of computing continuous functions. Its principal limitation in existing embodiments is its speed. However, it is possible at the present time to increase this speed by at least two orders of magnitude (from approximately 100 to 10,000 iterations per second). This will extend the DDA's range of usefulness to the point where it will not only be superior to the GP machine in more areas, but will also be fast enough to replace analog computers in many high-speed applications.

Mr. Glassman (Burroughs): The integration performed in a DDA is not real-time in the sense that it is in an analog computer, is it?

Rubinoff: This wasn't implied at all. It was a question of having the information arrive at the right point at the right time so that there is no wasted time in trying to arrange to get the information when you want it, as you do in a conventional digital computer. There are no instructions, for example, no housekeeping, no adjusting, none of that. Everything goes in a smooth, rapid manner.

Bauer: Real-time only means the effective speed at which computations are performed, not the method. Serial computation can be real-time, just as well as parallel computation. What is not true is that in the serial operation every variable in the machine, or its analog, has exactly the value it should have at every instant.

Braun: Integration in a DDA is performed in a different manner than in an analog machine, and it is not in "real time" in the sense that the time taken for the process is determined by the "real" response time of an electronic circuit. However, as generally used in computing circles, the term "real time" refers not to the manner but rather to the rate of a computation. What is usually meant when it is said that a computer operates in "real time" is that it operates sufficiently fast to keep up with a physical process being followed or controlled. Since different physical processes take place at different rates, the term "real time" is relative. Its use always implies reference to the rate at which a specific physical process occurs.

Glassman: I have one more question with regard to the DDA. When taking variables in from the outside world, you would like to have them, as I understand,

come into the DDA in an incremental manner. How do you accomplish this?

Braun: There was insufficient time to discuss the input-output problem in the oral presentation, though it is discussed in the written presentation. Therefore, I will only comment briefly here on how input variables are treated in a DDA.

It is not necessary to have external inputs enter a DDA in an incremental manner. In fact, inputs are usually entered very much the same as in a GP machine. Special registers, termed servo registers, are available in a DDA, and they can be used to generate a unitary weighted pulse train from absolute binary input data. The output of the servo register is the signum function of its contents. This output is also fed back to the input of the register with a sign reversal. Thus, whether the contents of the register is a positive or negative number, the register tends to drive its contents to zero. This occurs at the rate of a single increment per sampling cycle, and in the process a unitary weighted pulse stream is generated which more or less follows, on the average, the variations of the input variable.

Dr. Skramstad: One way to get variables into the DDA from an outside analog source would be to convert the analog input to digital form and transfer directly to a DDA register. Another way of doing this would be to produce an analog voltage proportional to the contents of a register, compare this with the analog voltage to be inserted, and use the difference to cause increments to be added or subtracted from the register until balance is obtained. It could be arranged so that whenever the rate of change of the variable exceeds the counting rate that the DDA can follow, the machine counts by 2's or 4's or at whatever speed is necessary in order to follow.

Mr. B. K. Smith (Beckman Instruments): The usual way of getting an analog input into a DDA is to perform a digital-to-analog conversion on a number representing that input, and then "update" that number according to the polarity of the output of an analog comparator.

I would like to ask whether anyone in the audience or panel knows of an incremental computer wherein the sampling rate is made a function of the rate of change; that is, where an input is observed to be frequently changing, but at much longer intervals in the normal scan sequence. It seems to me that a speed increase of, perhaps, a hundred-fold could be obtained in systems where only a few inputs vary at any given time.

Chairman Elbourn: Apparently no one knows of such a device.

Bauer: Some of us on the panel have referred to the so-called interrupt feature, and we mentioned that we are going to see a great deal more of this. As you know, it is a standard feature on the 1103 computer, and is also available on the 704 computer. On neither of these, nor on any computer that I know of, are multiple interrupts available.

Mr. Turner whispers in my ear that at N.A.C.A. they can distinguish interrupts, and that is the point I wish to make: you must not only have the interrupts, but you will need computer distinguishable interrupts, too, so that if three or four or more outside equipments happen to be interrupting, the computer will be able to distinguish among them.

The other point about the interrupt is this: we are seeing for the first time in digital computers the computer working on two problems simultaneously—on a time-sharing basis, of course. For instance, one idea we have in our shop is to supply, let us say, 100 cells for the analog computer to use just as it wishes. Then the analog computer, by means of the interrupt and the analog-digital converting equipment can steal time without the digital computer's knowing about it, so to speak. That is, the digital computer can be working on an entirely different problem, be interrupted at times by the analog computer, and therefore essentially do two problems simultaneously.

Mr. Turner: I would like to extend that comment a little. There is a point in connection with interrupt which must not be forgotten in making a computer solve two problems at one time: the base problem must be guaranteed to be free of errors. Actually, our second attempt at interrupt was to completely override a standby problem with provisions for returning to this problem later.

It gave rise to a rather amusing situation. One of our problems involved a computer-output typewriter. The man who had the control over the interrupt had a push-button. Whenever he pushed it, it would automatically interrupt the data processing problem. He managed one time to interrupt one line of typing four times.

Bauer: I don't know if it is proper for panel members to ask questions of other panel members, but I have a question to ask of Professor Susskind. He mentioned the fact that it was impossible, or at least very difficult, to put feedback resulting from measuring the effect of the cutting device back into the computer and thereby readjust the position of the cutting device. I wonder whether the loop can't be extended to make an approximate cut, go back over the cut, and measure and put the resulting measurement back into the computer. Following this there would be some calculations, and then another cut, and so forth. Thus one would close the loop in a longer fashion rather than directly as would first be conceived. This would involve more extensive computer operation, I am sure, but I wonder if it is feasible.

Mr. Susskind: I don't see any reason why this could not be done. I suspect the only limitation here is not technology but economics.

Mr. Wakefield: May I ask a question? In a system which includes general-purpose equipment and DDA's and analog equipment, how do you check on the output when it is required to be continuous? In a general-purpose equipment, you can stop the computation to

check the results. How do you do this in DDA or analog equipment?

Bauer: I am not quite sure whether I understand the question. I guess you check out the whole system in the same way as you would check out an analog computer where you have the same difficulty of stopping in the middle of the problem; that is, run it all the way through and see if the answers are reasonable. Of course, there are many static tests that can be placed on the equipment before it is tied together and run as a system.

Susskind: May I put in a comment here which slightly contradicts one of the other speakers? In machine-tool control one usually speaks of a data processing system which is small enough to be made at least as reliable as the basic machine tool. I am not at all worried about putting any amount of electronics on the machine-tool provided I can justify it economically, and I am not worried about downtime on the machine-tool or spoiling parts. We have probably cut in our lab, strictly on an experimental basis, 400 or 500 parts, and we spoiled only two of them. One was spoiled by the machine, the other by the operator.

Glassman: I would like to ask the speaker from Beckman a question. I think you indicated that the digital computer in a control loop could be checked by the analog servo controller. If the digital computer made an error, there would be some method to check by the analog control. I don't see how you can do that. Is that what you said or did I misunderstand you?

Gunning: That is what I said, and by that I meant to imply something along the lines of incremental data transmission—the sort of thing that you and the panel have been discussing. For example, assume that the set point of the servoregulator is allowed to move only a small increment each time the digital computer gives it an instruction. If the digital computer were to request a change that is very much larger than the maximum value that you know would be "reasonable" for the process, then the analog regulator would take the small step that is allowed to take and signal back to the digital computer that an unreasonable request has been made.

Mr. Weiss (Waldorf Instrument Co.): I have a question directed toward the gentleman who presented the paper in reference to the 15-channel A and D converter. Was that converter a separate unit from the 1103, or was it strictly fed into the 1103?

The second part of my question is, was it strictly a 15-channel input which is electro-input, or was it a combination electro-mechanical?

Bauer: The converter equipment is a tailor-made piece of equipment by the Epsco Company, of Cambridge, Mass. It is of very high speed, it is very accurate, and it is very expensive. It is completely electronic with speeds of 90 microseconds per analog-to-digital conversion, and it is made up of somewhat standard components and techniques, however, with some extra equipment, such as an oscillator for clock pulses and control mechanisms, which give it flexibility in use. It has fifteen channels of analog-to-digital conversion and fifteen channels of digital-to-analog conversion, giving it a 30-channel capacity. I am not sure I have answered all of your questions.

Weiss: The other half of my question was this. Among the input channels, do you have a system by which you feed in mechanical data also, as well as electrical data?

Bauer: The converter will, of course, read into the digital computer anything that can be presented on the plus or minus 100 volts scale of the analog computer. This means, for example, that we can take FM tape and play it through the proper equipment into the converter and, in turn, into the digital computer.

Glassman: In connection with that question, I don't understand why you are using 1-millivolt accuracy when 100 volts is what you get out of the converter. Why are you using such a converter as that?

Bauer: I believe that the 1-millivolt accuracy comes from the desire to measure and convert precisely those quantities which are small, as well as the quantities which have a large voltage value.

Question: Is it 1-millivolt accuracy or resolution?

Bauer: We refer to it as accuracy. I am not sure that I know the exact difference between them; I am sure there are various interpretations of those two words.

Rubinoff: I would like to say a little more about the original question that Dr. Bauer raised. I hesitate to start on it because it is a topic I have done quite a bit of thinking on and there is a danger of my overstepping available time. But I would like to give you some indications of the direction of the thinking.

We have been talking today about digital computers that are capable of doing more than one problem because they have an interrupt feature, about digital computers which are tied in with analog computers, and about digital computers which are tied in with DDA's. The question I would like to raise is, what is the proper size of module that we ought to be talking about? We seemed to be talking about the module at the level of a complete computer. And here we are talking again from the standpoint of a complete converter tied to a complete computer tied to another complete computer, one computer being digital, the other analog.

What little bit of thinking has been done so far indicates that maybe the size of module is wrong. Maybe because everybody said ten years ago that it is very difficult to build a digital computer, the feeling has grown that a complete computer is a basic module. Perhaps we have forgotten that that was something that was important ten years ago but not necessarily important today.

As I said, I don't want to go into a long oration, but I do want to put across this last thought, that perhaps a module ought to be just a storage register or a group of storage registers and another module just a counter, another module just an accumulator. Perhaps the whole should be set up in such a way that all of these are inter-

connected in such a manner as to make optimum use of the equipment.

This again ties in to the difference between a DDA and a digital computer. A DDA works much more efficiently on the job for which it was built because it is set up to do that job without a lot of housekeeping and getting prepared and shifting around of data. Even though it is inherently a slower, less costly machine, it competes favorably in many cases with a high-speed, expensive digital computer.

The question is: Can we take smaller modules and put them together in such a way that we keep these modules active more nearly 100 per cent of the time, rather than called into play perhaps 10 or 15 per cent of the time as is the current situation?

Mr. Flesch (Federal Telecommunication Laboratories): I would just like to remark on this last comment, that it seems that this would replace the program by a logical design for each problem.

Rubinoff: Not at all. As a matter of fact, the plan that we are working on is to eliminate altogether the man who schedules the problem for the computer. Why not just put in the priority information at the outset and let the whole system put itself together in any way it pleases to handle the problems as they come, in accordance with the priorities that are put on them? At present this sounds like a real dream, as digital computers may have sounded fifteen years ago, but, upon first examination at any rate, it looks as though this is quite feasible. There are, in fact, some really attractive features, and it is not nearly as expensive as one might at first think, because so much is saved on the digital computer equipment, the actual computation equipment, that you can afford to put it into the priority determination, of if you like, into the master control. I don't think it really makes life any more difficult for the programmer, particularly now that we have automatic programming.

Chairman Elbourn: Dr. Rubinoff, your remarks remind me a little bit of microprogramming, building the digital computer in such a way that the more elementary operations of the machine are available, such as a simple transfer from Register A to Register B, so that either by plugging or programming one can synthesize complex computer operations and possibly make a DDA out of a general-purpose computer if desired.

Rubinoff: Correct. Microprogramming is what I have in mind, but microprogramming in such a way that the system is always using all the equipment, and operating on several problems at one time. This apparently is not too difficult to mechanize. The problem is still an economic one, and it may be years before such a system can be achieved dollar-wise, but conceptually there is real hope in this area.

Braun: When I spoke of the relative advantages of absolute and incremental computation, I did not mean to imply that one should necessarily interconnect a conventional GP computer and a DDA. For some applications this might be adequate. However, in general, one would wish to look at the combination of absolute and incremental computation at the level of small modules in order to see logical relations that might allow simplification of equipment requirements. Storage registers, counters, and accumulators, as Dr. Rubinoff suggests, are very reasonable choices for modules. Such devices are basic for all common types of digital computation, that is, for absolute arithmetic units, for digital integrators, and for binary rate multipliers, as well as for control circuitry in general. However, the lower the level of integration, the greater the probability that one will have a computer that is relatively more specialized in function.

In regard to Dr. Rubinoff's remark about the speed of a DDA, I would like to suggest that the DDA is not inherently slower than a GP machine in the solution of differential equations. Current DDA's, operating in a completely serial mode with a magnetic drum store, may be slower than GP machines having considerable parallel circuitry and high-speed random access storage, but this is hardly a valid comparison. However, as Dr. Rubinoff points out, even these DDA's compete favorably with GP machines in many instances.

A few general comments may be in order on the subject of GP computers. There has been, in the past, a great emphasis on this type of machine. In the early stages of a new development, the general purpose approach is justifiable on the following grounds. First, the requirements of potential users usually are not clearly defined, and a GP machine provides a customer with more assurance that, even if his requirements change, he will not be left with a piece of equipment for which he will have relatively little use. Second, development costs can be spread out over a wider area. However, one does sacrifice efficiency for general utility. It is not unreasonable to expect that in the future, as large classes of special applications become more clearly defined, it will be economical to develop a number of types of special purpose computers, each of which is most efficient in the solution of a particular class of problems.

Mr. Wolf: I just want to comment on Mr. Gunning's remark on the reliability of a digital computer in a real-time industrial control system. While it may be uneconomical for that purpose, we, at Sage, use a very large number of both special-purpose and general-purpose interconnected digital computers, and, by means of redundancy—on a very large scale to be sure—we hope to have an over-all system of reliability which is characterized more by the six months than by one day. While this probably would not be economical for an industrial process right now, it does show that it is at least technically feasible to have a such control system even for systems that are very likely to be more complicated than any industrial system.

Chairman Elbourn: With reference to the reliability required in industrial control systems, I was a bit shaken one day by some visitors to the Bureau of Standards

from Imperial Chemical Industries, Ltd., of Great Britain. They wanted to know if any work is being done on digital computer circuitry that could be used for chemical industry process control. And they explained that they would not like to have a major breakdown any oftener than once in five years. Until then I thought we had already achieved rather good reliability; the question showed me how far we have yet to go. To do that kind of thing, we shall have to introduce the kind of system that Bill Gunning mentioned, namely, a hierarchy of control loops with perhaps unsophisticated but very reliable things in the lowest control loops, and the more sophisticated computing equipment superimposed in higher loops, in such a way that if something goes wrong there, it only affects the system about as much as one martini affects a human being.

Solid-state components certainly should let us make some of these lower control loops as reliable as the pneumatic equipment that is now being used.

Mr. Lipple (Signal Corps Engineering Laboratories): I was just wondering whether, as a result of some of these last comments, it wouldn't be well to point out that where at one time you had general-purpose machines, nowadays you tend to get different types of general-purpose machines; maybe the concept of a general-purpose machine, or the name, has outlived its usefulness. Maybe we should talk about multipurpose machines and realize that for different kinds of jobs we may have need in the future for different kinds of multipurpose machines.

A multipurpose machine for simulation, or for some of the real-time problems, might include many of the concepts that were discussed this morning.

Chairman Elbourn: We have come to the end of the time for this session. We certainly appreciate your participation.

Logical Machine Design: A Selected Bibliography*

DOUGLAS B. NETHERWOOD†

Summary—**This paper presents a selection of current bibliographic references in the field of the logical design of machines. Selections were made with the objective of being useful and stimulatory to the wide variety of scientific persons who have some interest in this subject. An essential feature of the report is the extensive index of significant title words. Each indexed word is given in the complete context of the title in which it occurs.**

Introduction

IN THE PREFACE to "Elements of Symbolic Logic," Professor Hans Reichenbach says, ". . . the logistic approach to philosophy is not bound to a certain type of mind, or of milieu or educational system, but represents a most successful clarification of ideas, by which all forms of scientific pursuit will profit." The design of finite automata is a scientific pursuit in which this observation appears to be sharply appropriate. The widely increasing emphasis now being given to research in mathematical logic as applied to computing machine networks and components is convincing evidence. The originality and usefulness of results already obtained should provide ample encouragement for continued endeavor along these lines.

The title of this report can be read in two ways, and it should be. The area of interest includes not only the use of mathematical logic in machine design (as contrasted with empirical, cut-and-try techniques) but also the use of logic machines to help solve difficult or lengthy problems in logic and even to generate new concepts and formulations. See, for example, [24].[1]

Scope

The reader may be surprised to find that Boole, despite the profound importance of his work (in mid-19th century), does not appear in this bibliography. Nor do Jevons or De Morgan. The investigations of such early writers have been thoroughly revised and assimilated into more modern works of wider application. This report does not offer a historical compilation but lists 1) informative papers on what has been done and is being done, and 2) stimulatory papers on what needs to be done and what might be possible.

Because the discoveries and techniques of logic have proved uniquely effective in the design and analysis of discrete-state machines, as opposed to analog or continuous-variable types, nearly all titles listed are pertinent to that subject. However, the area of interest has been somewhat broadened beyond the limits of the central subject alone (the direct use of logic in digital and finite-state problems). An attempt has been made to assemble a spectrum of material in which almost any investigator, regardless of specialized approach, can find

* Manuscript received by the PGEC, November 8, 1957.
† Electronic Components Lab., Wright Air Dev. Center, Wright-Patterson AF Base, Dayton, Ohio.

[1] Numbers in brackets throughout this report refer to bibliographic listings.

topics of interest. A particular effort has been made to anticipate and accommodate such individuals as, for example, an engineer who might be scanning the titles in this field and ask: "How have some of these abstract ideas been reduced to practice? What are some new phenomena and devices that offer further realizations?" On the other hand, a mathematician might say: "Given these system requirements and unsolved problems, what are some theoretical developments and studies of possible pertinence to their solution?"

The range of technical difficulty of the papers listed is from rigorous mathematical expositions to popular articles and news items, although these latter have been confined to a few notices of unusual promise. A few papers have been included from related areas such as machine philosophy, information theory, game theory, neurological analogies, and the mutual influence between programming and coding and computer logic and design.

Accessibility and readability of papers have been given some consideration, but the more significant titles have been listed regardless of probable availability or language of publication. An effort has been made to minimize the effect of personal bias on the part of the compiler. However, his opinion that truth-function theory and solid-state physics devices are especially promising is undoubtedly reflected in the selections.

Organization of the Bibliography and Index

Some thought was given to separating the bibliographic listings into categories. At least half of the titles could be so arranged, but the remainder would so overlap the boundaries that it would be necessary to make either multiple entries or arbitrary assignments. Therefore, a different procedure was selected.

Authors are given in alphabetical order in the Bibliography; but the Index is the key to this entire work. All significant title words appear as entries in the Index, together with the remainder of the title in each case. Where the significant word is not first in the title, a slant line precedes it. By this means, it is hoped that desired papers and related ones can be located with ease and accuracy. The reasoning for this arrangement is that most authors have selected their titles with care, and that no better method could be chosen than to give the salient words they have settled upon. The fairly common practice of listing single words in an index, followed by all relevant reference numbers, was rejected because it does not show the reader the context and may force him to search through dozens of unwanted references to find the one desired.

The use of the Index is straightforward. One has only to locate known or probable title words and note the associated reference numbers. In searching for titles involving phrases such as "Boolean algebra," "system design," or "logical matrix," it is recommended that both words be checked.

Reading Suggestions

For readers who may be unfamiliar with the literature in this field and who would like to look at some introductory material, it is suggested that some selections be taken from among the following references: [30], [51], [62], [107], [145], [195], [237], [243], [262], [301], [364], [384], [456].

Periodicals in which pertinent articles are frequently to be found, arranged in approximate sequence of technical difficulty, are as follows:

Electronic Design
Automatic Control
Computers and Automation
Instruments and Automation
Control Engineering
IRE Transactions on Electronic Computers
Journal of the Association for Computing Machinery
Journal of Computing Systems
Journal of Symbolic Logic.

Other Bibliographies

There are a number of excellent bibliographies in related fields. Some of the more important of these are:

1) Aiken, H. H., and Staff of the Computation Laboratory, Harvard University. "Bibliography of numerical analysis" in "A Manual of Operation for the Automatic Sequence Controlled Calculator," Harvard University Press, Cambridge, Mass., pp. 338–396; 1946.

2) Army Library. "Data Processing Operations and Computers for Management," Special Bibliography No. 10, Dept. of the Army; October 4, 1956.

3) Beach, A. F., *et al.* "Bibliography on the Use of IBM Machines in Science, Statistics, and Education," IBM Corp., New York, N. Y.; 1954.

4) Bureau of Labor Statistics. "Automatic Technology and its Implications; a Selected Annotated Bibliography," Dept. of Labor Bulletin No. 1198; August, 1956.

5) Chapin, N. "Publications for business on automatic computers: a basic listing," and " . . . a supplemental listing," *Computers and Automation*, vol. 4, September, 1955; vol. 5, February, 1956.

6) Chessin, P. L. "A bibliography on noise," IRE Trans. on Information Theory, vol. IT-1, pp. 15–31; September, 1955.

7) Church, A. "A bibliography of symbolic logic" (parts I and II), *J. Symbolic Logic*, vol. 1 (no. 4, entire) and vol. 3, pp. 178–212 (available in separate reprint); December, 1936, and December, 1938.

8) Clark, F. A. "Computers; Including Fire Control Computers, Flight Calculators, Mathematical Computers, Navigation Computers: A Report Bibliography," ASTIA Reference Center, Library of Congress, Report No. ARC-854.

9) Comstock, M. "Bibliography on Machine Computation 1945–1954," Brookhaven National Lab., ASTIA No. AD 62539; May, 1955.

10) Culbertson, J. T. "Robots and automata: a short history. bibliography," *Computers and Automation*, vol. 6, pp. 20–24, 38; April, 1957.

11) Hollander, G. L. "Bibliography on Data Storage and Recording," Servomechanisms Lab., M.I.T., Cambridge, Mass., ASTIA No. AD 15789; 1953. Also in *Trans. AIEE*, vol. 73, pp. 49–58; March, 1954.

12) McCann, G. D. "Bibliography of Articles on Computing Machines and their Applications," California Inst. Tech., Pasadena, Calif.; October, 1949.

13) Miles, J. G. "Bibliography of magnetic amplifier devices and the saturable reactor art," *Trans. AIEE*, vol. 70, pt. 2, pp. 2104–2117; 1951.

14) Nottingham, W. B. "Bibliography on Physical Electronics," Res. Lab. of Electronics, M.I.T., Cambridge, Mass.; 1954.

15) Parke, N. G., III, "Guide to the Literature of Mathematics and Physics Including Related Works on Engineering Science," McGraw-Hill Book Co., Inc., New York, N. Y.; 1947.

16) Serrell, R., and Rupprecht, K. J. "Bibliography on Electronic Computing," (3rd ed. rev.), RCA Labs. Div., Radio Corp. of America, Princeton, N. J.; July, 1953.

17) Stumpers, F. L. "A Bibliography on Information Theory," Report, Res. Lab. of Electronics, M.I.T., Cambridge, Mass.; February, 1953. Reprinted by IRE-PGIT, November, 1953. " . . . Supplement," IRE TRANS. ON INFORMATION THEORY, vol. IT-1, pp. 31–47; September, 1955.

18) Wade, W. F., and Kemler, E. N. "Automatic Control Bibliography," Summary Reports, Spring Park, Minn.; 1955.

ACKNOWLEDGMENT

This bibliography was prepared in the Advanced Development Branch of the Electronic Components Laboratory, Wright Air Development Center.

The assistance given by the following persons is gratefully acknowledged: K. D. Allen, O. H. Althoff, F. J. Anderson, J. R. Anderson, G. L. Burkhardt, H. Dubner, Miss E. Hollaway, Dr. F. R. Smith, Mrs. M. Vincze, Mrs. J. G. Williamson, and Miss G. A. Wolfe.

BIBLIOGRAPHY

[1] Aaron, M. R. "The use of least squares in system design," IRE TRANS. ON CIRCUIT THEORY, vol. CT-3, pp. 224–231; December, 1956.

[2] Abbott, W. R. "Computing logical truth with the California digital computer," *Math. Tables and Other Aids to Comp.*, vol. 5, pp. 120–128; 1951.

[3] Ackermann, W. "Solvable Cases of the Decision Problem," North-Holland Publishing Co., Amsterdam, The Netherlands; 1954.

[4] Aiken, H. H., and Staff of the Computation Laboratory, Harvard University. "Synthesis of Electronic Computing and Control Circuits," Harvard University Press, Cambridge, Mass.; 1951.

[5] Albers-Schoenberg, E. "Ferrites for microwave circuits and digital computers," *J. Appl. Phys.*, vol. 25, pp. 152–154; 1954.

[6] Albert, A. A. "Modern Higher Algebra," University of Chicago Press, Chicago, Ill.; 1937.

[7] Alonso, R., and Conley, T. "Ferroelectric applications to digital computers," *U. S. Govt. Res. Reps.*, vol. 26, p. 12; July, 1956.

[8] Alpert, N., *et al.* "32-channel high-speed commutator," *Electronics*, vol. 23, pp. 94–97; November, 1950.

[9] Anderson, A. E. "Transistors in switching circuits," PROC. IRE, vol. 40, pp. 1541–1558; November, 1952.

[10] Anderson, J. R. "Ferroelectric storage elements for digital computers and switching systems," *Elec. Eng.*, vol. 71, pp. 916–922; October, 1952.

[11] Andrews, L. J. "Fundamentals of inhibit-core logic," Los Angeles Chapter, IRE-PGEC, 11 pp.; December, 1956.

[12] Andrus, W. E., Jr. "The use of computing machines in the statistical evaluation of electrical components," *Commun. and Electronics*, pp. 501–505; November, 1956.

[13] Anonymous. "Automatic computing machinery—list of types," *Computers and Automation*, vol. 6, pp. 22–23, 25; March, 1957.

[14] ——. "Components of automatic computing machinery—list of types," *Computers and Automation*, vol. 6, pp. 24–25; March, 1957.

[15] ——. "Components and building blocks for computers," *Electronic Design*, vol. 3, pp. 64–69; October, 1955.

[16] ——. "The magic capsules," *Time*, pp. 79–80; May 13, 1957.

[17] ——. "The prodigies," *Time*, p. 83; March 19, 1956.

[18] ——. "Recent developments in computing," *Elec. J.*, vol. 156, pp. 917–919; March 23, 1956.

[19] ——. "Research and development trends, computers," *Aviation Age*, p. H-3; June, 1956; pp. I 3–I 11, Handbook; 1957.

[20] ——. "Solid synthesis," *Chem. Eng. News*, vol. 35, pp. 24–25; April 22, 1957.

[21] Aranovich, B. I. "The use of matrix methods in problems of the structural analysis of relay-contact networks" (Russian), *Avtom. Telemek.*, vol. 10, pp. 437–451; 1949.

[22] Ashby, W. R. "Design for a Brain," John Wiley and Sons, Inc., New York, N. Y.; 1952.

[23] ——. "Design for an intelligence-amplifier," *Automata Studies* (400), pp. 215–233.

[24] ——. "The effect of experience on a determinate dynamic system," *Behavioral Science*, vol. 1, pp. 35–42; January, 1956.

[25] Ashenhurst, R. L. "Functions Expressible as a Sum of Transforms," Bell Labs. Rep. No. 2, IV 1–IV 4; 1953.

[26] ——. "A Method for Synthesizing Multiple-output Networks Sequentially," Bell Labs. Rep. No. 1, V 1–V 6; 1953.

[27] Astrahan, M. M., and Rochester, N. "The logical organization of the new IBM scientific calculator," *Proc. Assoc. Comp. Mach.*; May, 1952.

[28] Astrahan, M. M., *et al.* "The logical design of a digital computer for a large-scale real-time application," *Proc. Western Joint Comp. Conf.*, pp. 70–75; February, 1956.

[29] Auerbach, I. L. "Fast-acting digital memory systems," *Elec. Mfg.*, pp. 100–107, 136–143; October–November, 1953.

[30] ——. "Static Magnetic Development," Burroughs Field Trip Rep., Tech. Memo. 52–62; June 6, 1952.

[31] Auerbach, I. L., and Disson, S. B. "Magnetic elements in arithmetic and control circuits," *Elec. Eng.*, vol. 74, pp. 766–770; September, 1955.

[32] Avion Division, ACF Industries, Inc. "Decision Procedures for the Functional Calculus," AF Cambridge Res. Center, Bedford, Mass., Rep. No. AFCRC-TN-56-387; May, 1956. Supplement: AFCRC-TN-56-785; September, 1956.

[33] ——. "Digitalized Logic and Its Applications," prepared by Data Processing Dept., Avion; September, 1955.

[34] ——. "Digitalization of the Propositional Calculus and the Monadic Functional Calculus," AF Cambridge Res. Center, Bedford, Mass., Rep. No. AFCRC-TN-56-176; February, 1956.

[35] ——. "Investigations of Formal Systems of Reasoning Leading to Machine Processing," AF Cambridge Res. Center, Bedford, Mass., Rep. No. AFCRC-TR-56-166; ASTIA Doc. No. AD 110103; November, 1956.

[36] ——. "Mechanization Techniques for Digitalized Logic," AF Cambridge Res. Center, Bedford, Mass., Rep. No. AFCRC-TN-56-593; August, 1956.

[37] Baker, H., *et al.* "Phase-bistable transistor circuit," PROC. IRE, vol. 41, pp. 1119–1124; September, 1953.

[38] Barnes, R. C. M., *et al.* "An electronic digital computer using cold-cathode counting tubes for storage," *Electronic Eng.*, vol. 23; pp. 286–291, 341–343; 1951.

[39] Belevitch, V. "Recent Russian publications on switching theory" (reviews), IRE TRANS. ON CIRCUIT THEORY, vol. CT-3, pp. 77–79; March, 1956.

[40] Bennett, A. A., and Baylis, C. A. "Formal Logic," Prentice-Hall, Inc., New York, N. Y.; 1939.

[41] Bennett, W. S. "Minimizing and mapping sequential circuits," *Trans. AIEE*, vol. 74, pt. 1, pp. 443–447; September, 1955.

[42] Berkeley, E. C. "The algebra of states and events," *Sci. Monthly*, vol. 78, pp. 232–242; April, 1954. Review [94].

[43] ——. "Circuit Algebra—Introduction," E. C. Berkeley and Associates, New York, N. Y., rev. ed.; August, 1955.

[44] ——. "Giant Brains; or Machines that Think," John Wiley and Sons, Inc., New York, N. Y.; 1949.

[45] ——. "The relations between symbolic logic and large-scale calculating machines," *Science*, vol. 112, pp. 395–399; 1950.

[46] ——. "A Summary of Symbolic Logic and Its Practical Applications," E. C. Berkeley and Associates, New York, N. Y.; June, 1954.

[47] Berkeley, E. C., and Wainwright, L. "Computers; Their Operations and Applications," Reinhold Publishing Corp., New York, N. Y.; 1956.

[48] Beter, R. H., *et al.* "Surface barrier transistor switching circuits," 1955 IRE CONVENTION RECORD, pt. 4, pp. 139–145.

[49] Bing, K. "On simplifying truth-functional formulas," *J. Symbolic Logic*, vol. 21, pp. 253–254; September, 1956.

[50] Birkhoff, G. "Lattice theory," in "Encyclopaedia Britannica," vol. 13, pp. 761–762; 1955.

[51] ——. "Lattice Theory," Amer. Math. Soc. Colloq. Publ., vol. 25; 1948.

[52] Birkhoff, G., and MacLane, S. "A Survey of Modern Algebra," The Macmillan Co., New York, N. Y.; 1947.
[53] Birkhoff, G. D., and Birkhoff, G. "Distributive postulates for systems like Boolean algebras," *Trans. Amer. Math. Soc.*, vol. 60, pp. 3–11; July, 1946.
[54] Booth, A. D. "On optimum relations between circuit elements and logical symbols in design of electronic calculators," *J. Brit. IRE*, vol. 12, pp. 587–594; 1952.
[55] ——. "Storage devices for high speed calculators," *Research*, vol. 8, pp. 130–140; April, 1955.
[56] Booth, A. D., and Booth, K. H. V. "Automatic Digital Calculators," Butterworths Scientific Publications, London, Eng.; 1953.
[57] Booth, G. W., and Bothwell, T. P. "Basic logic circuits for computer applications," *Electronics*, vol. 30, pp. 196–200; March, 1957.
[58] ——. "Logic circuits for a transistor digital computer," IRE Trans. on Electronic Computers, vol. EC-5; pp. 132–138; September, 1956.
[59] Bourbaki, N. "Structures Algébriques," ASEI 1141, Hermann and Cie, Paris, France, 2nd ed.; 1951.
[60] ——. "Théorie des Ensembles," Fascicule de Resultats ASEI 1141, Hermann and Cie, Paris, France, 2nd ed.; 1951.
[61] Bowden, B. V., ed. "Faster than Thought," Sir Isaac Pitman and Sons, London, Eng.; 1953.
[62] Bower, G. G. "Traveling-wave tube as a computer component," *U. S. Govt. Res. Reps.*, vol. 26, p. 329; December, 1956.
[63] Bowman, J. R. "Reduction of the number of possible Boolean functions," *Proc. Ninth Macy Conf.*, pp. 120–126; 1952.
[64] Bradshaw, T. F. "Automatic data processing methods," *Proc. Automatic Data Processing Conf.*, Harvard University Press, Cambridge, Mass.; September 8–9, 1955.
[65] Bright, R. L. "Junction transistors used as switches," *Trans. AIEE*, pt. 1, vol. 7, pp. 111–121; March, 1955.
[66] Brown, R. M. "Some notes on logical binary counters," IRE Trans. on Electronic Computers, vol. EC-4, pp. 67–69; June, 1955.
[67] Bruce, G. D., and Logue, J. C. "Experimental transistorized calculator," *Elec. Eng.*, vol. 74, pp. 1044–1048; December, 1955.
[68] Buchholz, W. "System design of the IBM type 701 computer," Proc. IRE, vol. 41, pp. 1262–1275; October, 1953.
[69] Buck, D. A. "The Cryotron—a superconductive computer component," Proc. IRE, vol. 44, pp. 482–493; April, 1956.
[70] ——. "Ferroelectrics for Digital Information Storage and Switching," Digital Computer Lab., M.I.T., Cambridge, Mass., Rep. R-212; June 5, 1952.
[71] Buffery, G. H. "A contribution to the algebra of relay and switch contacts," *Proc. IEE*, vol. 97, pt. 1, pp. 357–363; November, 1950.
[72] Burkhart, W. H. "Theorem minimization," *Proc. Assoc. Comp. Mach.*, pp. 259–263; May, 1952.
[73] Burks, A. W. "The logic of programming electronic digital computers," *Indust. Math.*, vol. 1, pp. 36–52; 1950.
[74] ——. "Super electronic computing machine," *Electronic Ind.*, vol. 5, pp. 62–67, 96; July, 1946.
[75] ——. "The logical design of an idealized general-purpose computer," *J. Franklin Inst.*, vol. 261, pp. 297–314, March, 1956; pp. 421–436, April, 1956.
[76] Burks, A. W., and Wright, J. B. "Theory of logical nets," Proc. IRE, vol. 41, pp. 1357–1365; October, 1953.
[77] Burks, A. W., *et al.* "An analysis of a logical machine using parenthesis-free notation," *Math. Tables and Other Aids to Comp.*, vol. 8, pp. 53–57; 1954. Review [299].
[78] ——. "Complete decoding nets: general theory and minimality," *J. Soc. Indust. Appl. Math.*, vol. 2, pp. 201–243; 1954. Review [300].
[79] ——. "The folded tree," *J. Franklin Inst.*, vol. 260, pp. 9–24, 115–126; 1955.
[80] Caldwell, S. H. "The recognition and identification of symmetric switching functions," *Trans. AIEE*, vol. 73, pt. 1, pp. 142–146; 1954. Review [301].
[81] Calhoun, E. S. "Computer competition from other continents," *Automatic Control*, vol. 6, pp. 62–68; April, 1957.
[82] Canning, R. G. "Electronic Data Processing for Business and Industry," John Wiley and Sons, New York, N. Y.; 1956.
[83] Carmichael, R. D. "Introduction to the Theory of Groups of Finite Order," Ginn and Co., Boston, Mass.; 1937.
[84] Carnap, R. "Foundations of logic and mathematics" in "International Encyclopedia of Unified Science," vol. 1, no. 3, University of Chicago Press, Chicago, Ill.; 1939.
[85] Carter, W. C., and Ellis, M. "A comparison of order structures for automatic digital computers," *J. Oper. Res. Soc. Amer.*, vol. 2, pp. 41–58; 1954.
[86] Carter, W. C., and Rettig, A. S. "Analytical minimization methods in conjunctive forms," *J. Comp. Systems*, vol. 1, pp. 179–195; July, 1953.
[87] Cavalieri, A. L. "What's inside Transac," *Electronic Design*, vol. 4, pp. 22–25, July 1, 1956; pp. 30–33, July 15, 1956.
[88] Chaplin, G. B. B. "The transistor regenerative amplifier as a computer element," *Proc. IEE*, vol. 101, pt. 3, pp. 298–307; 1954.
[89] Chase, S. "The logical animal," *J. Communication*, vol. 6, pp. 95–102; 1956.
[90] Cherry, E. C. "Generalized concepts of networks," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y., vol. 3, pp. 175–184; 1954.
[91] Chomsky, N. "Three models for the description of language," IRE Trans. on Information Theory, vol. IT-2, pp. 113–124; September, 1956.
[92] Church, A. "Introduction to Mathematical Logic," Princeton University Press, Princeton, N. J., vol. 1; 1956.
[93] ——. "A note on the Entscheidungsproblem," *J. Symbolic Logic*, vol. 1, pp. 40–41, 101–102; 1936.
[94] ——. Review of [42]. *J. Symbolic Logic*, vol. 20, pp. 286–287; September, 1955.
[95] ——. Review of [206]. *J. Symbolic Logic*, vol. 18, pp. 345–346; December, 1953.
[96] ——. Review of [212]. *J. Symbolic Logic*, vol. 18, pp. 347–348; December, 1953.
[97] ——. "Special cases of the decision problem," *Rev. Phil. de Louvain*, vol. 49, pp. 203–221; 1951.
[98] Chwistek, L. "The Limits of Science," Harcourt, Brace, and Co., New York, N. Y.; 1948.
[99] Clark, R. C. "Static design of transistor flip-flops and related circuits," Los Angeles Chapter, IRE-PGEC, 23 pp.; December, 1956.
[100] Collins, H. W. "Magnetic amplifier control of a switching transistor," *Commun. and Electronics*, no. 27, pp. 585–589; November, 1956.
[101] Clos, C. "A study of non-blocking switching networks," *Bell Sys. Tech. J.*, vol. 32, pp. 406–424; 1953.
[102] Cooley, J. C. "A Primer of Formal Logic," The Macmillan Co., New York, N. Y.; 1946.
[103] Cooper, J. L. B. "Mathematical monsters," *Math. Gazette*, vol. 38, pp. 258–265; 1954.
[104] Copi, I., and Harary, F. "Some properties of *N*-adic relations," *Portugal. Mathemat.*, vol. 12, pp. 143–152; 1953.
[105] Craven, T. L. "Logic and the circuit designer," *Electronic Eng.*, vol. 25, pp. 257–259; 1953.
[106] Cray, S. R., and Kisch, R. N. "A progress report on computer applications in computer design," *Proc. Western Joint Comp. Conf.*, pp. 82–85; February, 1956.
[107] Culbertson, J. T. "Hypothetical Robots," Rand Project P-296; April, 1952.
[108] ——. "Some uneconomical robots," *Automata Studies* (400), pp. 99–116.
[109] Curry, H. B. "The combinatory foundations of mathematical logic," *J. Symbolic Logic*, vol. 7, pp. 49–64; 1942.
[110] ——. "Mathematics, syntactics, and logic," *Mind*, vol. 62, pp. 172–183; 1953. Review [131].
[111] ——. "On the definition of substitution, replacement and allied notions in an abstract formal system," *Rev. Phil. de Louvain*, vol. 50, pp. 251–269; 1952. Review [131].
[112] ——. "A Theory of Formal Deducibility," Notre Dame Mathematical Lectures, No. 6; 1950.
[113] Dacey, G. C., and Ross, I. M. "The field effect transistor," *Bell Sys. Tech. J.*, vol. 34, pp. 1149–1189; November, 1955.
[114] ——. "Unipolar field effect transistor," Proc. IRE, vol. 41, pp. 970–979; August, 1953.
[115] Dantzig, G. B., and Fulkerson, D. R. "On the Max-flow Min-cut theorem of networks" in "Linear Inequalities (Annals of Math. Studies, No. 38)," Princeton University Press, Princeton, N. J.; 1956.
[116] David, E. E. "Ears for computers," *Sci. Amer.*, vol. 192, pp. 92–98; February, 1955.

[117] Davis, M. Review of [371]. *J. Symbolic Logic*, vol. 18, pp. 277–280; 1953.
[118] Davis, R. L. "The number of structures of finite relations," *Proc. Amer. Math. Soc.*, vol. 4, pp. 486–495; 1953.
[119] Dekker, J. C. E. "A theorem on hypersimple sets," *Proc. Amer. Math. Soc.*, vol. 5, pp. 791–796; 1954.
[120] Desoer, C. A. "The iterative solution of networks of resistors and ideal diodes," *Proc. Natl. Electronics Conf.*, vol. 2, pp. 678–685; 1955.
[121] Dietzold, R. L. "Network theory comes of age," *Elec. Eng.*, vol. 67, pp. 895–899; September, 1948.
[122] Dunham, B. "The multipurpose bias device. Part I. The commutator transistor," *IBM J. Res. Dev.*, vol. 1, pp. 116–129; April, 1957.
[123] Dunwell, S. W. "Design objectives for the IBM stretch computer," presented at Eastern Joint Comp. Conf., 1956.
[124] Eckert, J. P., Jr. "A survey of digital computer memory systems," Proc. IRE, vol. 41, pp. 1393–1406; October, 1953.
[125] Elias, P. "Error-free coding," IRE Trans. on Information Theory, no. PGIT-4, pp. 29–37; September, 1954.
[126] Elias, P., *et al.* "A note on the maximum flow through a network," IRE Trans. on Information Theory, vol. IT-2, pp. 117–119; December, 1956.
[127] Faris, J. A. "The Gergonne relations," *J. Symbolic Logic*, vol. 20, pp. 207–231; September, 1955.
[128] Farley, B. G., and Clark, W. A. "Simulation of self-organizing systems by digital computer," IRE Trans. on Information Theory, no. PGIT-4, pp. 76–84; September, 1954.
[129] Felker, J. H. "Arithmetic processes for digital computers," *Electronics*, vol. 26, pp. 150–155; March, 1953.
[130] ——. "Notes on the design of high speed digital computers using transistors" in "The Transistor," Bell Telephone Laboratories, pp. 627–751; 1951.
[131] Feys, R. Review of [110] and [111]. *J. Symbolic Logic*, vol. 21, pp. 375–377; December, 1956.
[132] Finney, J. W. "Transistor rival devised by Navy," (Solion), *N. Y. Times*, vol. 106, pp. 1, 38; June 23, 1957.
[133] Fletcher, T. J. "The solution of inferential problems by Boole algebra," *Math. Gazette*, vol. 36, pp. 183–188; 1952.
[134] Flood, J. E., and Warman, J. B. "The design of cold-cathode valve circuits III," *Electronic Eng.*, vol. 28, pp. 528–532; December, 1956.
[135] Forrester, J. W. "Digital information storage in three dimensions using magnetic cores," *J. Appl. Phys.*, vol. 22, pp. 44–48; January, 1951.
[136] Fowler, E. "Semi-conductor diode switches," *Field Engineers' Electronics Digest*, vol. 4, pp. 21–24; September, 1956.
[137] Frankel, S. P. "The logical design of a simple general purpose computer," IRE Trans. on Electronic Computers, vol. EC-6, pp. 5–14; March, 1957.
[138] Freeman, H. "Multipole sampled-data control systems," *U. S. Govt. Res. Reps.*, vol. 27, p. 56; February, 1957. ASTIA No. 81534.
[139] Frege, G. "The Foundations of Arithmetic," J. L. Austin, translator, Oxford University Press, Oxford, Eng.; 1950; Philosophical Library, New York, N. Y.; 1950.
[140] Fry, M. "Designing computing mechanisms," *Machine Design*, August, 1945 through February, 1946, *passim.*
[141] Fry, T. C. "The automatic computer in industry," *J. Amer. Statistical Assoc.*, vol. 51, pp. 565–575; December, 1956.
[142] Fuller, J. F. "New tricks for an old calculating board," *Genl. Elec. Rev.*, vol. 56, pp. 60; November, 1953.
[143] Gardner, M. "Logic machines," *Sci. Amer.*, vol. 186, pp. 68–73; March, 1952.
[144] Gardner, M. F., and Barnes, J. L. "Transients in Linear Systems," John Wiley and Sons, Inc., ch. 9; 1942.
[145] Gavrilov, M. A. "Relay schemes with vacuum-tube circuits," *Avtom. Telemek.*, vol. 16, pp. 328–343; 1955.
[146] General Electric Research Laboratory, "Glass and metal honeycomb type of electrostatic storage memory," *Computers and Automation*, vol. 5, pp. 10–11, 42; September, 1956.
[147] Ghazala, M. J. "Irredundant disjunctive and conjunctive forms of a Boolean function," *IBM J. Res. Dev.*, vol. 1, pp. 171–176; April, 1957.
[148] Gilbert, E. N. "Lattice theory properties of frontal switching circuits," *J. Math. Phys.*, vol. 33, pp. 57–67; April, 1954.
[149] ——. "*N*-terminal switching circuits," *Bell Sys. Tech. J.*, vol. 30, pp. 668–688; July, 1951.
[150] Gluck, S. E., *et al.* "Design of logical OR-AND-OR pyramids for digital computers," Proc. IRE, vol. 41, pp. 1388–1392; October, 1953.
[151] Golay, M. J. E. "Binary coding," IRE Trans. on Information Theory, no. PGIT-4, pp. 23–28; September, 1954.
[152] ——. "The logic of bidirectional binary counters," IRE Trans. on Electronic Computers, vol. EC-6, pp. 1–4; March, 1957.
[153] Goldstine, H. H., and Von Neumann, J. "Planning and Coding of Problems for an Electronic Computing Instrument," Institute for Advanced Study, Princeton, N. J., vol. 1, 1947; vol. 3, 1948.
[154] Goodell, J. D. "The foundations of computing machinery," *J. Comp. Systems*, vol. 1, pp. 1–13; 1952.
[155] ——. "The relations between logical, mathematical and computing machine systems," *J. Comp. Systems*, vol. 1, pp. 243–254; December, 1954.
[156] Grabbe, E. M. "Flight control and the digital computer," *Control Eng.*, vol. 2, pp. 64–68; October, 1955.
[157] Gray, H. J., Jr. "Logical description of some digital computer adders and counters," Proc. IRE, vol. 40, pp. 29–33; January, 1952.
[158] ——. "Numerical methods in real-time simulation," *Quart. Appl. Math.*, vol. 12, pp. 133–140; 1954.
[159] Grea, R., and Higonnet, R. "Etude logique de circuits de contact," *Rev. Gén. de l'Elect.*, vol. 38, pp. 19–34; 1954.
[160] Greenwald, S., *et al.* "SEAC," Proc. IRE, vol. 41, pp. 1300–1313; October, 1953.
[161] Grems, M., and Porter, R. E. "A truly automatic computing system," *Proc. Western Joint Comp. Conf.*, pp. 10–21; February, 1956.
[162] Greniewski, H., *et al.* "Applications of bi-elemental Boolean algebra to electronic circuits," *Studia Logica* (*Warsaw*), vol. 2, pp. 7–76; 1955. Review [302].
[163] Gruenberg, E. "Machines and religion," *Computers and Automation*, vol. 5, pp. 6–9, 45; January, 1956.
[164] Guillemin, E. A. "Mathematics of Circuit Analysis," John Wiley and Sons, Inc., New York, N. Y.; 1949.
[165] Guterman, S., Kodis, R. D., and Ruhman, S. "Logical and control functions performed with magnetic cores," Proc. IRE, vol. 43, pp. 291–298; March, 1955.
[166] Halmos, P. R. "Measure Theory," D. Van Nostrand Co., Inc., New York, N. Y.; 1950.
[167] ——. "Predicates, terms, operations, and equality in polyadic Boolean algebras," *Proc. Natl. Acad. Sciences, U.S.A.*, vol. 42, pp. 130–136; March, 1956.
[168] Ham, J. M. "A system-theory perspective for signal theory," IRE Trans. on Circuit Theory, vol. CT-3, pp. 208–209; December, 1956.
[169] Hamming, R. W. "Error detecting and error correcting codes," *Bell Sys. Tech. J.*, vol. 29, pp. 147–160; April, 1950.
[170] Harris, B. "An algorithm for determining minimal representation of a logic function," IRE Trans. on Electronic Computers, vol. EC-6, pp. 103–108; June, 1957.
[171] Harris, J. R. "Storage and addition of binary numbers," in "The Transistor," Bell Telephone Laboratories, pp. 523–581; 1951.
[172] ——. "Transistor shift register and serial adder," Proc. IRE, vol. 40, pp. 1597–1602; November, 1952.
[173] Hartree, D. R. "Calculating Instruments and Machines," University of Illinois Press, Urbana, Ill.; 1949.
[174] Harvard University Computation Laboratory. "Investigations for Design of Digital Calculating Machinery," Prog. Rep. 1 through 10, 13.
[175] Hayes, M. H., Jr. "Gate-tube Investigations," Servomechanisms Lab., M.I.T., Cambridge, Mass., NP 782; 1948.
[176] Henle, R. A. "A multistable transistor circuit," *Elec. Eng.*, vol. 74, pp. 570–572; July, 1955.
[177] Hilbert, D., and Ackermann, W. "Principles of Mathematical Logic," Chelsea Publishing Co., New York, N. Y.; 1950.
[178] Hobbs, G. W. "Selecting Matrix Study," General Electric Co., (2 vols.), U2892, 1948; U6642, 1949.
[179] Hoerl, A. E. "The large computer installation," *Chem. Eng. Prog.*, vol. 52, pp. 464–466; November, 1956.
[180] Hohn, F. E. "Some mathematical aspects of switching," *Amer. Math. Monthly*, vol. 62, pp. 75–90; 1955. Review [303].
[181] Hohn, F. E., and Schissler, L. R. "Boolean matrices and the design of combinational relay switching circuits," *Bell Sys. Tech. J.*, vol. 34, pp. 177–202; January, 1955. Review [329].

[182] Holt, A. W. "An Experimental Rapid Access Memory Using Diodes and Capacitors," *Natl. Bur. Standards*, AD 8778; 1953.
[183] Hopper, G. M., and Mauchly, J. W. "Influence of programming techniques on the design of computers," Proc. IRE, vol. 41, pp. 1250–1254; October, 1953.
[184] Hough, G. H., and Ridler, D. S. "Multicathode gas-tube counters," *Elec. Commun.*, vol. 27, pp. 214–226; September, 1950.
[185] Householder, A. S. "Generation of errors in digital computations," *Bull. Amer. Math. Soc.*, vol. 60, pp. 234–247; 1954.
[186] Huang, C., *et al.* "Field effect transistor applications," *Commun. and Electronics*, no. 25, pp. 323–329; July, 1956.
[187] Huffman, D. A. "The design and use of hazard-free switching networks," *J. Assoc. Comp. Mach.*, vol. 4, pp. 47–62; January, 1957.
[188] ——. "Information conservation and sequence transducers," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y., vol. 3, pp. 291–307; 1954.
[189] ——. "A Study of the Memory Requirements of Sequential Switching Circuits," Res. Lab. of Electronics, M.I.T., Cambridge, Mass., Tech. Rep. No. 293; April, 1955.
[190] ——. "The synthesis of sequential switching circuits," *J. Franklin Inst.*, vol. 257, pp. 161–190; March, 1954; pp. 275–303; April, 1954.
[191] Huggins, W. H. "Signal theory," IRE Trans. on Circuit Theory, vol. CT-3, pp. 210–216; December, 1956.
[192] Hunter, L. P., and Bauer, E. W. "High speed coincident-flux magnetic storage principles," *J. Appl. Phys.*, vol. 27, pp. 1257–1261; 1956.
[193] Huntington, E. V. "Fundamental Propositions of Algebra," Galois Institute of Mathematics and Art, Brooklyn, N. Y.; 1950.
[194] Huskey, H. D., *et al.* "The SWAC—design features and operating experience," Proc. IRE, vol. 41, pp. 1294–1299; October, 1953.
[195] Hussey, L. W. "Anticoincidence or 'NOT AND' gate," in "The Transistor," Bell Telephone Laboratories, pp. 517–522; 1951.
[196] ——. "Diode gates," in "The Transistor," Bell Telephone Laboratories, pp. 507–515; 1951.
[197] Hutchinson, G. E. "Marginalia," *Amer. Scientist*, vol. 43, pp. 144–147; January, 1955.
[198] Ivall, T. E. "Electronic Computers," Philosophical Library, New York, N. Y.; 1957.
[199] Jacobson, A. "Electronic computer is built of unit-packaged circuits," *Elec. Eng.*, vol. 73, p. 191; February, 1954.
[200] Jeffress, L. A., ed. "Cerebral mechanisms in behavior," John Wiley and Sons, Inc., New York, N. Y.; 1951.
[201] Jeffrey, R. C. "Arithmetical analysis of digital computing nets," *J. Assoc. Comp. Mach.*, vol. 3, pp. 360–375; October, 1956.
[202] Kalicki, J. "Note on truth-tables," *J. Symbolic Logic*, vol. 15, pp. 174–181; 1950.
[203] ——. "A test for the equality of truth-tables," *J. Symbolic Logic*, vol. 17, pp. 161–163; 1952.
[204] ——. "A test for the existence of tautologies according to many-valued truth-tables," *J. Symbolic Logic*, vol. 15, pp. 182–184; 1950.
[205] ——. "An undecidable problem in the algebra of truth-tables," *J. Symbolic Logic*, vol. 19, pp. 172–176; 1954.
[206] Kalin, T. A. "Formal logic and switching circuits," *Proc. Assoc. Comp. Mach.*, pp. 251–257; May, 1952. Review [95].
[207] Karnaugh, M. "The map method for synthesis of combinational logic circuits," *Trans. AIEE*, vol. 72, pt. 1, pp. 593–599; November, 1953.
[208] ——. "Pulse switching circuits using magnetic cores," Proc. IRE, vol. 43, pp. 570–583; May, 1955.
[209] Katz, D. "A magnetic amplifier switching matrix," *Commun. and Electronics*, no. 24, pp. 236–241; May, 1956.
[210] Kavanaugh, R. J. "The application of matrix methods to multi-variable control systems," *J. Franklin Inst.*, vol. 262, pp. 349–367; November, 1956.
[211] Kazan, B., and Knoll, M. "Fundamental processes in charge-controlled storage tubes," *RCA Rev.*, vol. 12, pp. 702–753; 1951.
[212] Keister, W., *et al.* "The Design of Switching Circuits," D. Van Nostrand Co., Inc., New York, N. Y.; 1951. Review [96].
[213] Kelley, R. "Magnetic-core registers," *Field Engineers' Electronics Digest*, vol. 4, pp. 32–34; September, 1956.
[214] ——. "Magnetron beam switching tubes," *Field Engineers' Electronics Digest*, vol. 4, pp. 7–9; September, 1956.
[215] Kelner, R. C. "Components needed for digital computers," *Electronic Design*, vol. 3, pp. 32–33; October, 1955.
[216] Kemeny, J. G. "Man viewed as a machine," *Sci. Amer.*, vol. 192, pp. 58–67; April, 1955.
[217] King, G. W. "A new approach to information storage," *Control Eng.*, pp. 48–53; August, 1955.
[218] Klass, P. J. "Solid state advances, trends detailed," *Aviation Week*, vol. 66, pp. 101–109; May 13, 1957.
[219] ——. "Solid-state research brews upheaval," *Aviation Week*, vol. 66, pp. 86–94; April 8, 1957.
[220] Kleene, S. C. "Introduction to Metamathematics," D. Van Nostrand Co., Inc., New York, N. Y.; 1952.
[221] ——. "Representation of events in nerve nets and finite automata," *Automata Studies* (400), pp. 3–41.
[222] ——. Review of [438]. *J. Symbolic Logic*, vol. 20, p. 74; March, 1955.
[223] Klein, M. L. "Simplified method for design of logical conversion matrices," *Electronic Eng.*, vol. 27, pp. 270–272; June, 1955.
[224] Kochen, M. "Group behavior of robots," *Computers and Automation*, vol. 6, pp. 16–21, 48; March, 1957.
[225] ——. "Organized systems with discrete information transfer," *J. Assoc. Comp. Mach.* (forthcoming).
[226] Kochen, M., and Levy, M. J., Jr. "Logical nature of an action scheme," *Behavioral Science*, vol. 1, pp. 265–289; October, 1956.
[227] Kodis, R. D. "Using magnetic cores in computers," *Electronic Design*, pp. 21–25; April, 1955.
[228] König, D. "Theorie der Endlichen und Unendlichen Graphen," Akad. Verlagsgesellschaft, Leipzig, Germany; 1936.
[229] Koppel, H. "Digital computer plays NIM," *Electronics*, vol. 25, pp. 155–157; November, 1952.
[230] Kornfield, N. R. "Pulsed Magnetic Core Devices," Burroughs Corp., Philadelphia, Pa., Special Rep. No. 1, Contract DA-49-170-sc-1033; 1953.
[231] Kuchinsky, S. "Beam-switching tubes for computers and controls," *Tele-Tech*, vol. 12, pp. 88–90+; September, 1953.
[232] Kudlich, R. A. "A set of transistor circuits for asynchronous, direct-coupled computers," *Proc. Western Joint Comp. Conf.*, pp. 124–129; March, 1955.
[233] Laemmel, A. E. "Design of digital coding networks," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y., vol. 3, pp. 309–320; 1954.
[234] Langer, S. K. "An Introduction to Symbolic Logic," Dover Press, New York, N. Y., 2nd rev. ed.; 1953.
[235] Ledley, R. S. "Mathematical foundations and computational methods for a digital logic machine," *J. Oper. Res. Soc. Amer.*, vol. 2, pp. 249–274; 1954. Review [304].
[236] Lee, C. Y. "Analysis of switching networks," *Bell Sys. Tech. J.*, vol. 34, pp. 1287–1315; 1955.
[237] ——. "Switching functions on an N-dimensional cube," *Trans. AIEE*, vol. 73, pt. I, pp. 289–291; September, 1954. Review [301].
[238] Lefshetz, S. "Algebraic Topology," Amer. Math. Soc. Colloq. Publ., vol. 27; 1942.
[239] Leichner, G. H. "A fast circulating memory," IRE Trans. on Electronic Computers, vol. EC-6, pp. 123–124; June, 1957.
[240] Lerner, I. S. "Digital computers need orderly number systems," *Control Eng.*, vol. 2, pp. 82–89; November, 1955.
[241] Lewis, C. I., and Langford, C. H. "Symbolic Logic," Dover Press, New York, N. Y.; 1932.
[242] Lewis, I. A. D. "A symbolic method for the solution of some switching and relay-circuit problems," *Proc. IEE*, vol. 98, pt. 1, pp. 181–191; May, 1951.
[243] Lieber, L. R., and Lieber, H. G. "Mits, Wits, and Logic," W. W. Norton and Co., Inc., New York, N. Y.; 1947.
[244] ——. "THE great discovery of modern mathematics," IRE Trans. on Military Electronics, vol. MIL-1, pp. 10–18; March, 1957.
[245] Lincoln, A. J. "Ferromagnetic Core Logical Circuitry and its Application to Digital Computers," Ballistic Research Labs., Aberdeen Proving Ground, Md., Memo Rep. No. 911, Dept. of Commerce OTS No. PB 111954.
[246] Linsman, M. "Sur une méthode de simplification de circuits

dans la théorie des fonctions de commutation," *Bull. Soc. Roy. Sci. Liège*, vol. 25, pp. 163–166; 1956.
[247] Linvill, W. K. "System theory as an extension of circuit theory," IRE Trans. on Circuit Theory, vol. CT-3, pp. 217–223; December, 1956.
[248] Lo, A. W. "Transistor trigger circuits," Proc. IRE, vol. 40, pp. 1531–1540; November, 1952.
[249] Locke, W. N., and Booth, A. D. "Machine Translation of Languages," John Wiley and Sons, Inc., New York, N. Y.; 1955.
[250] Lode, T. "The realization of a universal decision element," *J. Comp. Systems*, vol. 1, pp. 14–22; June, 1952.
[251] Loebner, E. E. "Opto-electronic devices and networks," Proc. IRE, vol. 43, pp. 1896–1906; December, 1955.
[252] Loev, D., *et al.* "Magnetic core circuits for digital data-processing systems," Proc. IRE, vol. 44, pp. 154–162; February, 1956.
[253] Lorenzen, P. "Einführung in die Operative Logik und Mathematik," Lange and Springer, Berlin, Germany; 1955.
[254] Lunts, A. G. "The application of Boolean matrix algebra to the analysis and synthesis of relay-contact networks," (Russian). *Dokl. Akad. Nauk SSSR*, vol. 70, pp. 421–423; 1950.
[255] Lyndon, R. C. "Identities in two-valued calculi," *Trans. Amer. Math. Soc.*, vol. 71, pp. 457–465; 1951. Review [445].
[256] MacDonald, N. D. "The project stretch computer (IBM)," *Computers and Automation*, vol. 6, pp. 8–9; April, 1957.
[257] MacKay, D. M. "Comparing the brain with machines," *Amer. Scientist*, vol. 42, pp. 261–268; April, 1954.
[258] ——. "The epistemological problem for automata," *Automata Studies* (400), pp. 235–251.
[259] ——. "Quantal aspects of scientific information," IRE Trans. on Information Theory, London Symp. on Information Theory, pp. 60–80; February, 1953.
[260] MacWilliams, W. H., Jr. "Computers; past, present, and future," *Electronic Eng.*, vol. 72, pp. 116–121; 1953.
[261] Marcus, M. P. "Minimization of the partially-developed transfer tree," IRE Trans. on Electronic Computers, vol. EC-6, pp. 92–95; June, 1957.
[262] Maron, M. E. "Logic, discovery, and the foundations of computing machinery," IRE Trans. on Electronic Computers, vol. EC-3, pp. 2–7; June, 1954.
[263] Marshall, B. O., *et al.* "Optical Elements for Computers," Mellon Inst. of Industrial Research, University of Pittsburgh, Pittsburgh, Pa., Quart. Rep. No. 6, Computer Components Fellowship No. 347; June, 1952.
[264] Mason, S. J. "Topological analysis of linear nonreciprocal networks," Proc. IRE, vol. 45, pp. 829–838; June, 1957.
[265] Mays, W., *et al.* "Note on the exhibition of logical machines at the joint session, July, 1950," *Mind*, vol. 60; pp. 262–264; 1951.
[266] McCallum, D. M., and Smith, J. B. "Feedback logical computers," *Electronic Eng.*, vol. 23, pp. 458–461; December, 1951.
[267] ——. "Mechanized reasoning—logical computers and their design," *Electronic Eng.*, vol. 23, pp. 126–133; 1951.
[268] McCluskey, E. J., Jr. "Detection of group invariance or total symmetry of a Boolean function," *Bell Sys. Tech. J.*, vol. 35, pp. 1445–1453; November, 1956.
[269] ——. "Minimization of Boolean functions," *Bell Sys. Tech. J.*, vol. 35, pp. 1417–1444; November, 1956.
[270] McCracken, D. D. "The Monte Carlo method," *Sci. Amer.*, vol. 192, pp. 90–96; May, 1955.
[271] ——. "Word length in digital computers," *Computers and Automation*, vol. 5, pp. 14–15; July, 1956.
[272] McCulloch, W. S. "The brain as a computing machine," *Elec. Eng.*, vol. 68, pp. 492–497; 1949.
[273] ——. "Machines that Think and Want," Comparative Psychology Monographs 20, pp. 39–50; 1950.
[274] McCulloch, W. S., and Pitts, W. "How we know universals," *Bull. Math. Biophysics*, vol. 9, pp. 127–147; 1947.
[275] ——. "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophysics*, vol. 5, pp. 115–133; 1943.
[276] McMahon, R. E. "Designing transistor flip-flops," *Electronic Design*, vol. 3, pp. 24–27; October, 1955.
[277] McNaughton, R. Review of [311] and [345]. *J. Symbolic Logic*, vol. 21, pp. 328–330; September, 1956.
[278] ——. Review of [312]. *J. Symbolic Logic*, vol. 21, pp. 330–331; September, 1956.
[279] Meisling, T. H. "Translation of arithmetic operations into switching operations in digital computer design," *U. S. Govt. Res. Reps.*, vol. 25, pp. 290–291; June, 1956.
[280] Miller, G. A. "Human memory and the storage of information," IRE Trans. on Information Theory, vol. IT-2, pp. 129–137; September, 1956.
[281] Milnor, J. "Sums of Games," Rand Corp. RM-654; July 27, 1951.
[282] Minnick, R. C., and Ashenhurst, R. L. "Multiple coincidence magnetic storage systems," *J. Appl. Phys.*, vol. 26, pp. 575–577; May, 1955.
[283] Minsky, M. L. "Some universal elements for finite automata," *Automata Studies* (400), pp. 117–128.
[284] Mock, O. R. "Logical organization of the PACT I compiler," *J. Assoc. Comp. Mach.*, vol. 3, pp. 279–287; October, 1956.
[285] Montgomerie, G. A. "Sketch for an algebra of relay and contactor circuits," *J. IEE*, vol. 95, pp. 303–312; 1948. Review [305].
[286] Moore, E. F. "Gedanken-experiments on sequential machines," *Automata Studies* (400), pp. 129–153.
[287] ——. "A simplified universal turing machine," *Proc. Assoc. Comp. Mach.*, September, 1952.
[288] Moore, E. F., and Shannon, C. E. "Reliable circuits using less reliable relays. Parts I and II," *J. Franklin Inst.*, vol. 262, pp. 191–208, September, 1956; pp. 281–297, October, 1956.
[289] Mueller, R. K. "On the Synthesis of a Minimal Representation of a Logic Function," Air Force Cambridge Res. Center, Bedford, Mass., Tech. Rep. No. 55-104; April, 1955.
[290] Muller, D. E. "Application of Boolean algebra to switching circuit design and to error detection," IRE Trans. on Electronic Computers, vol. EC-3, pp. 6–12; March, 1954. Review [306].
[291] Murnaghan, F. D. "The Theory of Group Representations," Johns Hopkins Press, Baltimore, Md.; 1938.
[292] Murphy, R. W. "A positive-integer arithmetic for data processing," *IBM J. Res. Dev.*, vol. 1, pp. 158–170; April, 1957.
[293] Murray, F. J. "Mechanisms and robots," *J. Assoc. Comp. Mach.*, vol. 2, pp. 61–82; April, 1955.
[294] ——. "Theory of Mathematical Machines," King's Crown Press, New York, N. Y., rev. ed.; 1948.
[295] Myhill, J. "On the interpretation of the Sign '$\supset$'," *J. Symbolic Logic*, vol. 18, pp. 60–62; March, 1953.
[296] National Bureau of Standards. "Diode amplifier, semiconductor amplifier offers important applications in computers and other electronic devices," *Tech. News Bull.*, vol. 38, pp. 145–148; 1954.
[297] Nelson, E. C. "An algebraic theory for use in digital computer design," IRE Trans. on Electronic Computers, vol. EC-3, pp. 12–21; September, 1954.
[298] ——. "Digital computers need logical design," *Control Eng.*, vol. 2, pp. 60–66; December, 1955.
[299] Nelson, R. J. Review of [77]. *J. Symbolic Logic*, vol. 20, pp. 70–71; March, 1955.
[300] ——. Review of [78]. *J. Symbolic Logic*, vol. 21, p. 210; June, 1956.
[301] ——. Review of [79] and [237]. *J. Symbolic Logic*, vol. 20, p. 197; June, 1955.
[302] ——. Review of [162]. *J. Symbolic Logic*, vol. 21, pp. 333–334; September, 1956.
[303] ——. Review of [180]. *J. Symbolic Logic*, vol. 21, pp. 209–210; June, 1956.
[304] ——. Review of [235]. *J. Symbolic Logic*, vol. 20, pp. 195–197; June, 1955.
[305] ——. Review of [285]. *J. Symbolic Logic*, vol. 20, pp. 68–69; March, 1955.
[306] ——. Review of [290]. *J. Symbolic Logic*, vol. 20, p. 195; June, 1955.
[307] ——. Review of [320]. *J. Symbolic Logic*, vol. 21, pp. 210–211; June, 1956.
[308] ——. Review of [342]. *J. Symbolic Logic*, vol. 18, pp. 280–282; September, 1953.
[309] ——. Review of [395]. *J. Symbolic Logic*, vol. 20, p. 69; March, 1955.

[310] ——. Review of [405]. *J. Symbolic Logic*, vol. 20, p. 70; March, 1955.
[311] ——. "Simplest normal truth functions," *J. Symbolic Logic*, vol. 20, pp. 105–108; June, 1955. Review [277].
[312] ——. "Weak simplest normal truth functions," *J. Symbolic Logic*, vol. 20, pp. 232–234; September, 1955. Review [278].
[313] Netherwood, D. B. "Logic Matrices," (in preparation).
[314] Newell, A., and Simon, H. A. "The logic theory machine," IRE Trans. on Information Theory, vol. IT-2, pp. 61–79; September, 1956.
[315] Newhouse, V. L. "A review of magnetic and ferroelectric computing components," *Electronic Eng.*, vol. 26, pp. 192–199; May, 1954.
[316] Newhouse, V. L., and Prywes, N. S. "High-speed shift register using one core per bit," IRE Trans. on Electronic Computers, vol. EC-5, pp. 114–120; September, 1956.
[317] Nicod, J. "Foundations of Geometry and Induction," The Humanities Press, New York, N. Y.; 1950.
[318] Noe, J. D. "Data processing systems: how they function," *Control Eng.*, vol. 2, pp. 70–77; October, 1955.
[319] Okada, S. "Algebraic and topological foundations of network synthesis," *Proc. Symp. on Network Synthesis*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y.; April, 1955.
[320] ——. "Topology applied to switching circuits," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y., pp. 267–290; 1954. Review [307].
[321] Olson, H. F., and Belar, H. "Phonetic typewriter," *J. Acous. Soc. Amer.*; November, 1956.
[322] Orey, S. Review of [412]. *J. Symbolic Logic*, vol. 21, pp. 322–323; September, 1956.
[323] Otis, E. J. "Series-Tube Core Memory," *U. S. Govt. Res. Reps.*, vol. 27, p. 14; January, 1957.
[324] Pap, A. "Strict implication, entailment, and modal iteration," *Phil. Rev.*, vol. 64, pp. 604–613; 1955.
[325] Papian, W. N. "New ferrite-core memory uses pulse transformers," *Electronics*, vol. 28, pp. 194–197; March, 1955.
[326] Parry, W. T. "A new symbolism for the propositional calculus," *J. Symbolic Logic*, vol. 19, pp. 161–168; 1954.
[327] Patterson, G. W. "Logical Syntax and Transformation Rules," University of Pennsylvania, Philadelphia, Pa., Res. Div. Rep. 50-8; 1949.
[328] ——. "Symbolic methods in the design of delay- and cycle-free logical nets," 1954 IRE Convention Record, pt. 4, pp. 58–64.
[329] Pawlak, Z. Review of [181]. *J. Symbolic Logic*, vol. 20, pp. 104–105; March, 1956.
[330] Penfield, P., Jr. "Information Theory," *Radio-Electronics*, pp. 96–101; April, 1956.
[331] Percival, W. S. "The graphs of active networks," *Proc. IEE*, vol. 102, pt. C, pp. 270–279; September, 1955.
[332] Petrick, S. R. "A Direct Determination of the Irredundant Forms of a Boolean Function from the Set of Prime Implicants," AF Cambridge Res. Center, Bedford, Mass., Tech. Rep. No. 56-110; April, 1956.
[333] Polya, G. "Mathematics and Plausible Reasoning," Princeton University Press, Princeton, N. J., 1954.
[334] ——. "Patterns of Plausible Inference," Princeton University Press, Princeton, N. J.; 1954.
[335] ——. "Sur les types des propositions composées," *J. Symbolic Logic*, vol. 5, pp. 98–103; 1940.
[336] Post, E. L. "Introduction to a general theory of elementary propositions," *Amer. J. Math.*, vol. 43, pp. 163–185; 1921.
[337] Postley, J. A. "A method for the evaluation of a system of Boolean algebraic equations," *Math. Tables and Other Aids to Computation*, vol. 9, pp. 5–8; 1955.
[338] Pulvari, C. F. "Memory matrix using ferroelectric condensers as bistable elements," *J. Assoc. Comp. Mach.*, vol. 2, pp. 169–185; July, 1955.
[339] Quine, W. V. "Logic based on inclusion and abstraction," *J. Symbolic Logic*, vol. 1, pp. 145–152; 1936.
[340] ——. "Mathematical Logic," Harvard University Press, Cambridge, Mass.; 1951.
[341] ——. "On ordered pairs," *J. Symbolic Logic*, vol. 10, pp. 95–96; 1945.
[342] ——. "The problem of simplifying truth functions," *Amer. Math. Monthly*, vol. 59; pp. 521–531; 1952. Review [308].
[343] ——. "A theory of classes presupposing no canons of type," *Proc. Natl. Acad. Sciences*, vol. 22, pp. 320–326; 1936.
[344] ——. "Unification of universes in set theory," *J. Symbolic Logic*, vol. 21, pp. 267–279; September, 1956.
[345] ——. "A way to simplify truth functions," *Amer. Math. Monthly*, vol. 62, pp. 627–631; 1955. Review [277].
[346] Raffel, J., and Bradspies, A. "Experiments on a three-core cell for high-speed memories," 1955 IRE Convention Record, pt. 4, pp. 64–69.
[347] Rajchman, J. A. "Ferrite apertured plate for random access memory," Proc. IRE, vol. 45, pp. 325–334; March, 1957.
[348] ——. "A myriabit magnetic-core matrix memory," Proc. IRE, vol. 41, pp. 1407–1421; October, 1953.
[349] ——. "Static magnetic matrix memory and switching circuits," *RCA Rev.*, pp. 183–201; June, 1951.
[350] Rajchman, J. A., and Lo, A. W. "The Transfluxor," Proc. IRE, vol. 44, pp. 321–332; March, 1956.
[351] Ramamoorthy, C. V., and Soroka, W. W. "Resistance wire computing device for solving algebraic equations," *Product Eng.*, vol. 23, pp. 134–137; February, 1952.
[352] Rapoport, A. "Random nets with transivity bias," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y.; vol. 3, pp. 187–197; 1954.
[353] Ratz, A. G., and Smith, V. G. "A method of gating for parallel computers," *Trans. AIEE*, vol. 70, pt. 1, pp. 510–516; 1951.
[354] Reed, I. S. "A class of multiple-error-correcting codes and the decoding scheme," IRE Trans. on Information Theory, pp. 38–49; September, 1954.
[355] ——. "Some Queue Processes Associated with Buffer-Storage Systems," Lincoln Lab., M.I.T., Lexington, Mass., Tech. Rep. No. 73; December, 1954.
[356] Reed, I. S., and Jeffrey, R. C. "The Symbolic Design of Digital Computers," Lincoln Lab. M.I.T., Lexington, Mass., Tech. Memo. No. 23; 1953.
[357] Reichenbach, H. "Elements of Symbolic Logic," The Macmillan Co., New York, N. Y.; 1947.
[358] Reiser, O. L. "Logic, Cybernetics, and General Semantics," General Semantics Bulletin No. 16, 17 (single issue); 1955.
[359] Respanti, M. "A complex algebra for relay circuits," *Elec. Eng.*, vol. 72, pp. 992–993; 1953.
[360] Richards, P. I. "On game-learning machines," *Sci. Monthly*, vol. 74, pp. 191–200; 1952.
[361] Richards, R. K. "Arithmetic Operations in Digital Computers," D. Van Nostrand Co., Inc., New York, N. Y.; 1955.
[362] Ridenour, L. N. "Computer memories," *Sci. Amer.*, vol. 192, pp. 92–100; June, 1955.
[363] ——. "High speed digital computers," *J. Appl. Phys.*, vol. 21, pp. 263–270; April, 1950.
[364] ——. "Role of the computer," *Sci. Amer.*, vol. 187, pp. 116–118; September, 1952.
[365] Riguet, J. "Sur le rapport entre les concepts de machine de multipole et de structure algébrique," *C. R. Acad. Sci.*, vol. 237, pp. 425–427; 1953.
[366] Riordan, J., and Shannon, C. E. "The number of two-terminal series parallel networks," *J. Math. Phys.*, vol. 21, pp. 83–93; 1942.
[367] Robbins, L. C. "An analysis by arithmetical methods of a calculating network with feedback," *Proc. ACM Toronto Meeting;* September, 1952.
[368] Robertson, J. H. "Simple machine capable of Fourier synthesis calculation," *J. Sci. Instr.*, vol. 27, pp. 276–278; 1950.
[369] Rochester, N., *et al.* "Tests on a cell assembly theory of the action of the brain, using a large digital computer," IRE Trans. on Information Theory, vol. IT-2, pp. 80–93; September, 1956.
[370] Rose, G. F. Review of [205]. *J. Symbolic Logic*, vol. 20, pp. 283–284; September, 1955.
[371] Rosenbloom, P. C. "The Elements of Mathematical Logic," Dover Press, 1950. Review [117].
[372] Rosenheim, D. E., and Anderson, A. G. "VHF pulse techniques and logical circuitry," Proc. IRE, vol. 45, pp. 212–219; February, 1957.
[373] Ross, H. D., Jr. "The arithmetic element of the IBM type 701 computer," Proc. IRE, vol. 41, pp. 1287–1294; October, 1953.
[374] Rosser, J. B. "Logic for Mathematicians," The McGraw-Hill Book Co., Inc., New York, N. Y.; 1953.
[375] Rothstein, J. "Information, organization and systems," IRE Trans. on Information Theory, pp. 64–75; September, 1954.
[376] Rudin, B. D. "A theorem on SPDT switching circuits," *Proc. Western Joint Comp. Conf.*, pp. 129–132; March, 1955.

[377] Rutz, R. F., and Berger, A. W. "A New Transistor with Thyratron-like Characteristics," IBM Tech. Bull. No. 101, presented at AIEE-IRE Conference, Los Angeles, Calif.; May 26, 1955.
[378] Samson, E. W., and Mills, B. E. "Circuit Minimization: Algebra and Algorithms for New Boolean Canonical Expressions," AF Cambridge Res. Center, Bedford, Mass., Tech. Rep. No. 54-21; April, 1954.
[379] Samson, E. W., and Mueller, R. K. "Circuit Minimization: Minimal and Irredundant Boolean Sums by Alternative Set Method," AF Cambridge Res. Center, Bedford, Mass., Tech. Rep. No. 55-109; June, 1955.
[380] ——. "Circuit Minimization: Sum to One Process for Irredundant Sums," AF Cambridge Res. Center, Bedford, Mass., Tech. Rep. No. 55-118; August, 1955.
[381] Samuel, A. L. "Computing bit by bit; or digital computers made easy," Proc. IRE, vol. 41, pp. 1223–1230; October, 1953.
[382] Schaefer, D. H. "A rectifier algebra," *Trans. AIEE*, vol. 74, pt. 1, pp. 679–682; 1955.
[383] Schmitt, W. F. "Single-core binary counter," (U. S. Patent No. 2,713,675), *Radio-Electronics*, p. 117; April, 1956.
[384] Scobey, J. E., *et al.* "Fast switching with junction diodes," Proc. IRE, vol. 44, pp. 1880–1881; December, 1956.
[385] Searles, H. L. "Logic and Scientific Methods," The Ronald Press Co., New York, N. Y., 2nd ed.; 1956.
[386] Semon, W. L. "Adding Transformations," Bell Labs. Rep. No. 4, I 1–I 9; 1953.
[387] ——. "Circuit Matrices," Bell Labs. Rep. No. 5, VIII 1–VIII 59; 1953.
[388] Serrell, R. "Elements of Boolean algebra for the study of information-handling systems," Proc. IRE, vol. 41, pp. 1366–1380; October, 1953.
[389] Shannon, C. E. "Computers and automata," Proc. IRE, vol. 41, 1234–1241; October, 1953.
[390] ——. "General treatment of the problem of coding," IRE Trans. on Information Theory, London Symposium on Information Theory, pp. 102–104; February, 1953.
[391] ——. "The lattice theory of information," IRE Trans. on Information Theory, London Symposium on Information Theory, pp. 105–107; February, 1953.
[392] ——. "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379–423; 623–656; 1948.
[393] ——. "Presentation of a maze-solving machine," *Trans. Eighth Cybernetics Conf., Macy Foundation*, pp. 173–180; 1952.
[394] ——. "A symbolic analysis of relay and switching circuits," *Trans. AIEE*, vol. 57, pp. 713–723; 1938.
[395] ——. "The synthesis of two-terminal switching circuits," *Bell Sys. Tech. J.*, vol. 28, pp. 59–98; January, 1949. Review [309].
[396] ——. "A universal Turing machine with two internal states," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y., vol. 3, pp. 157–165; 1954.
[397] Shannon, C. E., and McCarthy, J., eds. "Automata Studies," Annals of Math Studies No. 34, Princeton University Press, Princeton, N. J.; 1956. Review [481].
[398] Shannon, C. E., and Moore, E. F. "Machine aid for switching circuit design," Proc. IRE, vol. 41, pp. 1348–1351; October, 1953.
[399] Shannon, C. E., and Weaver, W. "The Mathematical Theory of Communication," University of Illinois Press, Urbana, Ill.; 1949.
[400] Shimbel, A. "Application of matrix algebra to communication nets," *Bull. Math. Biophysics*, vol. 13, pp. 165–178; 1941.
[401] ——. "Structure in communication nets," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y., vol. 3, pp. 199–203; 1954.
[402] Shockley, W. "A unipolar field effect transistor," Proc. IRE, vol. 40, pp. 1365–1376; November, 1952.
[403] Simkins, Q. W., and Vogelsong, J. H. "Transistor amplifiers for use in a digital computer," Proc. IRE, pp. 43–55; January, 1956.
[404] Slepian, D. "A class of binary signaling alphabets," *Bell Sys. Tech. J.*, vol. 35, pp. 203–234; January, 1956.
[405] ——. "On the number of symmetry types of Boolean functions of *N* variables," *Canad. J. Math.*, vol. 5, pp. 185–193; 1953. Review [310].
[406] Sluckin, W. "Mind and Machines," Penguin Books Ltd., London, Eng., and Baltimore, Md.; 1954.
[407] Smith, E. J. "Logical approach to the design of computer circuits," *Proc. Symp. on Information Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N. Y.; vol. 3, pp. 249–266; 1954.
[408] Smith, N. M., Jr. "A calculus for ethics: a theory of the structure of value," *Behavioral Science*, vol. 1, pp. 111–142, April, 1956; pp. 186–211, June, 1956.
[409] Snyder, R. L. "Magnistor circuits," *Electronic Design*, vol. 3, pp. 24–27; August, 1955.
[410] Stabler, E. R. "Boolean representation theory," *Amer. Math. Monthly*, vol. 51, pp. 129–132; 1944.
[411] Standley, G. B. "Ideographic computation in the propositional calculus," *J. Symbolic Logic*, vol. 19, pp. 169–171; 1954.
[412] Stanley, R. L. "Simplified foundations for mathematical logic," *J. Symbolic Logic*, vol. 20, pp. 123–139; June, 1955. Review [322].
[413] Statz, H., and Pucel, R. A. "The Spacistor, a new class of high-frequency semiconductor devices," Proc. IRE, vol. 45, pp. 317–324; March, 1957.
[414] Sternglass, E. J. and Wachtel, M. M. "Transmission secondary electron multiplication for high-speed pulse counting," IRE Trans. on Nuclear Science, vol. NS-3, pp. 29–32; November, 1956.
[415] Stibitz, G. R. "A note on 'is' and 'might be' in computers," *Math. Tables and Other Aids to Comp.*, vol. 4, pp. 168–169; 1950.
[416] Stone, M. H. "Postulates for Boolean algebras and generalized Boolean algebras," *Amer. J. Math.*, vol. 57, pp. 703–732; 1935.
[417] ——. "Representations of Boolean algebras," *Trans. Amer. Math. Soc.*, vol. 40, pp. 37–111; July, 1936.
[418] Storer, T. "The notion of 'tautology'," *Phil. Studies*, vol. 5; pp. 75–78; 1954.
[419] Svenonius, L. "Definability and simplicity," *J. Symbolic Logic*, vol. 20, pp. 235–250; September, 1955.
[420] Symposium. "The design of machines to simulate the human brain," IRE Trans. on Electronic Computers, vol. EC-5, 240–255; December, 1956.
[421] Symposium on The Theory of Switching. Computation Lab., Harvard University (in press).
[422] Sziklai, G. C., and Hill, L. R. "Electronic commutators by means of minority carrier deflection in semiconductors," presented at Symposium on the Role of Solid State Phenomena in Electric Circuits, Polytechnic Institute of Brooklyn, Brooklyn, N. Y.; April, 1957.
[423] Tarski, A. "A Decision Method for Elementary Algebra and Geometry," University of California Press, Berkeley, Calif., 2nd ed., rev.; 1951.
[424] ——. "Introduction to Logic and to the Methodology of Deductive Sciences," Oxford University Press, New York, N. Y.; 1941.
[425] Taylor, B. R. "The gas filled diode as a digital storage element," *Electronic Eng.*, vol. 28, pp. 151–155; April, 1956.
[426] Thompson, F. B. "Behavior Strategies in Finite Games," Rand Corp. RM-769; January 22, 1952.
[427] ——. "Equivalence of Games in Extensive Form," Rand Corp. RM-759; January, 1952.
[428] Toon, A. B., Jr. "Appraisal of electronic progress," *Price Waterhouse Rev.*, pp. 16–25; March, 1956.
[429] Trent, H. M. "Isomorphisms between oriented linear graphs and lumped physical systems," *J. Acous. Soc. Amer.*, vol. 27, pp. 500–527; May, 1955.
[430] Trent, R. L. "Four-stage binary counter," in "The Transistor," Bell Telephone Laboratories, pp. 501–505; 1951.
[431] ——. "A stabilized general purpose two-transistor binary counter," *ibid.*, pp. 469–482.
[432] ——. "Stabilized single transistor binary counter," *ibid.*, pp. 464–468.
[433] ——. "A transistor packaged circuit for multifunctional switching applications," *ibid.*, pp. 437–462.
[434] ——. "A transistor reversible binary counter," *ibid.*, pp. 483–500.
[435] Tung, C. C. "Diode coincidence and mixing circuits in digital computers," Proc. IRE, vol. 38, pp. 511–514; May, 1950.
[436] Turing, A. "On computable numbers with applications to the Entscheidungsproblem," *Proc. London Math. Soc.*, ser. 2, vol. 42, pp. 230–265; 1936.
[437] ——. "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460; 1950.
[438] ——. "Solvable and unsolvable problems," *Science News*, no. 31, pp. 7–23. Penguin Books, Ltd., London, Eng., and Baltimore, Md.; 1954. Review [222].

[439] Urbano, R. H., and Mueller, R. K. "A topological method for the determination of the minimal forms of a Boolean function," IRE Trans. on Electronic Computers, vol. EC-5, pp. 126–132; September, 1956.
[440] Uttley, A. M. "Conditional probability machines and conditional reflexes," *Automata Studies* (400), pp. 253–275.
[441] ——. "Information, machines, and brains," IRE Trans. on Information Theory, London Symposium on Information Theory, pp. 143–149; February, 1953. Discussion, pp. 193–197.
[442] ——. "Temporal and spatial patterns in a conditional probability machine," *Automata Studies* (400), pp. 277–285.
[443] Van der Poel, W. L. "A simple electronic digital computer," *Appl. Sci. Res.*, vol. 26, pp. 367–400; 1952.
[444] Van Nice, R. I. "Magnetic logic circuit control system design considerations," *Commun. and Electronics*, no. 27, pp. 595–600; November, 1956.
[445] Vaughan, H. E. Review of [255]. *J. Symbolic Logic*, vol. 18, pp. 69–70; March, 1953.
[446] Veblen, O. "Analysis Situs," Amer. Math. Soc. Colloq. Publ., vol. 5, pt. 2; 1931.
[447] Veitch, E. W. "A chart method for simplifying truth functions," *Proc. Assoc. Comp. Mach.*, pp. 127–133; May, 1952.
[448] Von Neumann, J. "The general and logical theory of automata," in "Cerebral Mechanisms in Behavior," John Wiley and Sons, Inc., New York, N. Y.; 1951.
[449] ——. "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Studies* (400), pp. 43–98.
[450] Von Neumann, J., and Morgenstern, O. "Theory of Games and Economic Behavior," Princeton University Press, Princeton, N. J.; 1947.
[451] Wall, R. E., Jr. "Some of the engineering aspects of the machine translation of language," *Commun. and Electronics*, pp. 580–585; November, 1956.
[452] Wallace, R. L., Jr., and Pietenpol, W. L. "Some circuit properties and applications of *n-p-n* transistors," *Bell Sys. Tech. J.*, vol. 30, pp. 530–563; July, 1951.
[453] Walter, W. G. "Possible features of brain function and their imitation," IRE Trans. on Information Theory, London Symposium on Information Theory, pp. 134–136; February, 1953.
[454] Wang, H. "Logic of many-sorted theories," *J. Symbolic Logic*, vol. 17, pp. 105–116; June, 1952.
[455] ——. "A variant to Turing's theory of computing machines," *J. Assoc. Comp. Mach.*, vol. 4, pp. 63–92; January, 1957.
[456] Wanlass, C. L. "Transistor circuitry for digital computers," IRE Trans. on Electronic Computers, vol. EC-4, pp. 11–16; March, 1955.
[457] Ware, W. H. "Logical principles of a new kind of binary counter," Proc. IRE, vol. 41, pp. 1429–1437; October, 1953.
[458] Washburn, S. H. "An application of Boolean algebra to the design of electronic switching circuits," *Trans. AIEE*, vol. 72, pt. 1, pp. 380–388; September, 1953.
[459] ——. "Relay 'trees' and symmetric circuits," *Trans. AIEE*, vol. 68, pt. 1, pp. 582–586; 1949.
[460] Weik, M. H. "Survey of domestic electronic digital computing systems," *U. S. Govt. Res. Reps.*, vol. 25, p. 107; March, 1956.
461] Weyl, H. "Philosophy of Mathematics and Natural Science," Princeton University Press, Princeton, N. J.; 1949.
[462] Whitehead, A. N., and Russell, B. "Principia Mathematica," Cambridge University Press, Cambridge, Eng., 2nd ed.; 1925.
[463] Whitney, G. E. "Magnetic scale-of-two" (U. S. Patent No. 2,710,928), *Radio-Electronics*, pp. 128–129; February, 1956.
[464] Wiener, N. "Cybernetics," John Wiley and Sons, Inc., New York, N. Y.; 1948.
[465] Wier, M. J. "Some direct-coupled computer circuits utilizing *NPN* and *PNP* transistors in combination," *U. S. Govt. Res. Reps.*, vol. 26, p. 329; December, 1956.
[466] Wilder, R. L. "Foundations of Mathematics," John Wiley and Sons, Inc., New York, N. Y.; 1952.
[467] Wittgenstein, L. "Tractatus Logico-Philosophicus," Kegan Paul, London, Eng.; 1922.
[468] Wolfendale, E., *et al.* "The junction transistor as a computing element," *Electronic Eng.*, vol. 29, pp. 2–7, January, 1957; pp. 83–87, February, 1957; pp. 136–139, March, 1957.
[469] Wong, S. Y., and Kochen, M. "Automatic network analysis with a digital computation system," *Commun. and Electronics*, vol. 24, pp. 172–176; May, 1956.
[470] Yokelson, B. J., and Ulrich, W. "Engineering multistage diode logic circuits," *Commun. and Electronics*, vol. 20, pp. 466–475; September, 1955.
[471] Young, J. W. "Fundamental Concepts of Algebra and Geometry," The Macmillan Co., Inc., New York, N. Y.; 1920.
[472] Zaheb, C. "Synthesis of 2-terminal switching circuits," *Commun. and Electronics*, p. 690; January, 1955.
[473] Zwicky, F. "On a new type of reasoning and some of its possible consequences," *Phys. Rev.*, vol. 43, pp. 1031–1033; 1933.

Addenda

[474] Rutz, R. F. "Two-collector transistor for binary full addition," *IBM J. Res. Dev.*, vol. 1, pp. 212–222; July, 1957.
[475] Shockley, W. "The four-layer diode," *Electronic Ind. & Tele-Tech*, vol. 16, pp. 58–60, 161–165; August, 1957.
[476] Wilkes, M. V. "Automatic Digital Computers," John Wiley and Sons, Inc., New York, N. Y.; 1956.
[477] Harary, F. "The number of linear, directed, rooted, and connected graphs," *Trans. Amer. Math. Soc.*, vol. 78, pp. 445–463; March, 1955.
[478] M.I.T. Research Laboratory of Electronics. "Three of von Neumann's biological questions," *Quart. Prog. Rep.*, pp. 129–138; October 15, 1957.
[479] Russell, L. A. "Diodeless magnetic core logical circuits," 1957 IRE National Convention Record, pt. 4, pp. 106–114.
[480] Abbott, H. W., and Suran, J. J. "Multihole ferrite core configurations and applications," Proc. IRE, vol. 45, pp. 1081–1093; August, 1957.
[481] Gilbert, E. N. Review of [397]. *J. Assoc. Comp. Mach.*, vol. 3, pp. 376–378; October, 1956.
[482] Rowe, W. D. "The transistor NOR circuit," 1957 IRE WESCON Convention Record, pt. 4, pp. 231–245.
[483] O'Toole, J. B. "Logic design symbolism for direct-coupled transistor circuits in digital computers," *ibid.*, pp. 251–258.
[484] Ellis, D. "A mathematical formulation of the generalized logical design problem," *ibid.*, pp. 259–261.

Index

(Numbers refer to listings in the Bibliography. Where there are two numbers, the second refers to a review of the original paper. A slant line precedes the first word in a title when some other word is cited.)

A

B

M

N

O

P

T

U

V

W

Correspondence

Unit-Distance Error-Checking Codes*

An interesting set of binary codes can be generated which combines the properties of *error checking*, such as single-error detection, and *unit distance*, which means that successive code elements (n-bit numbers) of the code sequence differ in only one bit. For example, the code sequence for $n=4$

0000
0001
0011
0111
1111
1110
1100
1000
(0000)

is of this type, in the sense that each single binary error in an element of the sequence results in either 1) the next element, 2) the preceding element, or 3) an element which does not appear in the sequence at all, and which may therefore indicate a "detected" error.

In this note some possible applications and a few properties of these codes are discussed, and several as yet unresolved problems and conjectures are offered for their possible interest.

These codes are potentially useful in two areas. First, they provide a type of error-detection code for digitalized analog data, in the sense that the most likely binary errors are either detected or result in negligible analog errors, and the analog-to-digital conversion may be accomplished with no commutation error. Second, in asynchronous digital networks without race conditions, they provide a code for counting which enables most simple circuit faults to be detected. Thus, if a failure in the circuitry causes a flip-flop to fail to change state when it should, or to change state when it shouldn't (or a relay to open or close inappropriately), then either the count will be in error by *one*, or, what is more likely for large n, the counter will advance to a forbidden state and the error arising from the fault may be detected. The price paid for this feature is an increased amount of logical circuitry in the counter. Such an arrangement is a type of "single-fault-detecting" sequential net, and is significant from a reliability standpoint.

The usual model of an error-checking code represents each of the code elements ($x_n x_{n-1} \cdots x_2 x_1$) as a vertex (node) of the n-dimensional unit cube[1] with coordinate directions $x_1, x_2 \cdots x_{n-1}, x_n$. A set of code elements with the single-error-detecting property is then represented by a set of vertices, no two of which are adjacent, *i.e.*, joined by a single edge of the cube. A code sequence with the unit-distance property is represented by a chain of adjacent vertices.[2] It is of interest in the design of codes having properties such as these to determine the maximum number of elements (nodes) which may be incorporated into the codes (cube), and to derive if possible a systematic pattern for generating such maximal-length codes for arbitrary n. The present problem is therefore one of packing the longest possible chain into the n cube with the restriction that no two nonsuccessive nodes on the chain be adjacent. (The interpretation of this chain as a unit-radius tube led to the term "snake in the box," or SIB, code.) One may consider open chains (segments) or closed chains (cycles); however, only closed chains are discussed here.

Let $K(n)$ be the maximum possible length of an SIB chain in the n cube. The following values have been derived, principally by trial and error:

n	2	3	4	5	6	7
$K(n)$	4	6	8	14	24	≥ 46.

The law of growth is not known, but the above sample, and a familiarity with the process of deriving the code in $n+1$ dimensions from the code in n dimensions, suggests the rule

$$K(n+1) = \begin{Bmatrix} 2K(n) - 2 \text{ for } n \text{ even} \\ 2K(n) - 4 \text{ for } n \text{ odd} \end{Bmatrix} \quad n \geq 2.$$

This recurrence relation leads to

$$K(n) = \tfrac{1}{3}[2^n - (-1)^n] + 3 \quad n \geq 2.$$

The following bounds are easily established:

$$K(n) \geq 2n$$

$$K(n) \leq \frac{n}{n-1} 2^{n-1}$$

$$K(n) \geq A 2^{n/2} \text{ for some constant } A \text{ independent of } n;\ A \geq 4.$$

The first of these bounds follows from the pattern of the above example for $n=4$. The second results from a count of the total number of edges in the cube, both on and off of the SIB path. The third results from the realization that an SIB path of length $2K(n)$ can be formed in $n+2$ dimensions from two SIB paths each of length $K(n)$, in n dimensions,[3] *i.e.*,

$$K(n+2) \geq 2K(n).$$

These bounds are generally rather poor, although the third one confirms the exponential growth displayed in the above conjectured formula for $K(n)$.

An alternate description of the structure of a unit-distance code sequence is provided by an ordered list of the coordinate changes required in passing from each element (node) to the next. For example, for the above $n=4$ SIB code with elements (x_4, x_3, x_2, x_1), we have, listing only subscripts:

$$n = 4\text{: } 12341234.$$

The values of $K(n)$ listed above are achievable with the following codes:

$n = 2$: 12 12*
$n = 3$: 123 123*
$n = 4$: 1234 1234*
1234 1243
1234 2143
$n = 5$: 1234524 123524*
1234524 1234254
1234523 1243253
$n = 6$: 123456423453 123456423453*
$n = 7$: 12345674534214561241534
21451675234563216346154

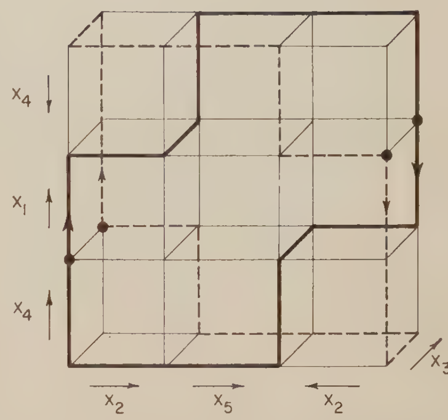


Fig. 1—The principal (solid line) and alternate (dashed line) natural SIB chains for $n=5$. $K(5)=14$.

For $n \leq 5$ the codes listed exhaust the symmetry types, *i.e.*, all other maximal-length SIB codes for $n \leq 5$ may be obtained from those listed by a reversal of direction along the path, a permutation of variables, choice of a different starting point, or some combination of these transformations. The number of symmetry types for $n \geq 5$ is not known. The starred (*) codes may be referred to as "natural" codes, in that they have the following unique special properties: 1) The path is antisymmetric; that is, the first and second halves of the path, from any starting point, follow the same coordinate-change sequence. 2) The reverse path can be derived from the forward path by merely inverting the order of the variables. 3) $K(n)$ of the $2^n - K(n)$ non-SIB-path nodes on the n cube may be selected (for $N>3$) for an alternate SIB path of the same symmetry type (see Fig. 1). 4) Less precisely, the logic circuitry required in conjunction with n flip-flops or relays arranged to count with this code sequence seems to be simpler, by almost any reasonable criterion, than for the

* Received by the PGEC, February 14, 1958.

[1] R. W. Hamming, "Error-detecting and error-correcting codes," *Bell Sys. Tech. J.*, vol. 29, pp. 147–160; April, 1950.

[2] H. E. Tompkins, "Unit-distance binary codes for two-track commutation," IRE Trans. on Electronic Computers, vol. EC-5, p. 139; September, 1956.

[3] J. Brenner of Stanford Res. Inst. reports a slightly better bound: $K(n) \geq B(2.25)^{n/2}$.

other symmetry types. The equations for the code of Fig. 1 are listed in Table I; here Z is the next value of z, Y the next value of y, etc. We conjecture that at least one such "natural" SIB code exists for arbitrary n.

TABLE I
EQUATIONS FOR $n=5$ SIB COUNTER WITH PRINCIPAL AND ALTERNATE PATHS

$$
\begin{aligned}
Z &= z[w + \bar{x} + y] + \bar{w}\bar{x}\bar{y} \\
Y &= y[\bar{w} + (x \oplus v)] + \bar{w}(x \oplus z) \\
X &= x[\bar{v} + z + (w \oplus \bar{y})] + \bar{v}z(w \oplus y) \\
W &= w[y + (x \oplus z)] + y(x \oplus v) \\
V &= v[\bar{w} + x + \bar{y}] + wxy \\
a \oplus b &\overset{\Delta}{=} a\bar{b} + \bar{a}b
\end{aligned}
$$

The SIB codes discussed above may be directly generalized to provide for multiple error checking, and to p-nary rather than binary codes.

The author is indebted to Bernard Elspas of Stanford Research Institute for his interest and participation in the study of these codes, and also benefited from contact with E. N. Gilbert of Bell Telephone Laboratories and Joel Brenner of Stanford Research Institute.

W. H. KAUTZ
Stanford Res. Inst.
Menlo Park, Calif.

An Optimum Character Recognition System Using Decision Functions*

Issue is taken with this statement from the above paper by Chow:[1] "The above results also indicate that the recognition system described in Section II (correlation coefficient) is not optimum for the particular signal and noise structure as given in examples 1 or 2."

This statement *may* be true generally but it is not true for a most important case. That is, for the case to be cited, correlation is an optimum recognition system.

Consider the following weights and probabilities:

$$
\begin{aligned}
W_{ii} &= 0 \qquad W_{i0} = 0 \qquad \text{for all } i \\
W_{ij} &= W \qquad \text{all } i \neq j \\
p_i &= p \qquad \text{all } i.
\end{aligned}
$$

This is the case when all characters are equally likely. It is equally costly to misidentify them, but costs nothing to identify them, and rejection is not considered.

The detector proposed examines the output

$$X_j(V) = \sum_i W_{ij} p_i F(V/a_i),$$

but

$$
\begin{aligned}
W_{ij} p_i &= 0 \qquad \text{for } i = j \\
&= Wp \qquad \text{for } i \neq j \\
X_j(V) &= Wp \sum_{i \neq j} F(V/a_i) \\
&= Wp\left(\sum_i F(V/a_i) - F(V/a_j) \right).
\end{aligned}
$$

To find the minimum X_j simply means to find max $F(V/a_j)$.

Examine $F(V/a_i)$ as given in (29)[1]

$$F(V/a_i) = \frac{\exp - \sum \left(\dfrac{(V_j - a_{ij})^2}{2\sigma_{ij}^2} \right)}{(2\pi)^{s/2} \Pi \sigma_{ij}}.$$

The hypothesis that the noise is independent means that the covariance matrix, $\|\sigma_{ij}^2\|$, is diagonal. Then $\sigma_{ij}=0$ for $i \neq j$. Call $\sigma_j = \sigma_{jj}$. Also, call $K = [(2\pi)^{s/2} \Pi \sigma_j]^{-1}$.

Now define Λ_i as follows, where Λ_i is the likelihood ratio discussed in reference 5 of Chow's paper:

$$\Lambda_i = F(V/a_i)/F(V)$$

where now

$$F(V/a_i) = K \exp\left(- \sum_j \frac{(V_j - a_{ij})^2}{2\sigma_j^2} \right)$$

and

$$F(V) = K \exp\left(- \sum_j V_j^2 / 2\sigma_j^2 \right).$$

Maximizing Λ_i corresponds to maximizing $F(V/a_i)$ as does maximizing log Λ_i. Therefore, examine

$$
\begin{aligned}
\log \Lambda_i &= \log F(V/a_i) - \log F(V) \\
&= \log K - \sum_j \frac{(V_j - a_{ij})^2}{2\sigma_j^2} - \log K \\
&\quad + \sum_j V_j^2 / 2\sigma_j^2, \\
-2 \log \Lambda_i &= \sum \frac{(V_j - a_{ij})^2}{\sigma_j^2} - \sum \frac{V_j^2}{\sigma_j^2} \\
&= \sum \frac{V_j^2 - 2a_{ij}V_j + a_{ij}^2}{\sigma_j^2} \\
&\quad - \sum V_j^2/\sigma_j^2 \\
&= \sum V_j^2/\sigma_j^2 - 2\sum a_{ij}V_j/\sigma_j^2 \\
&\quad + \sum a_{ij}^2/\sigma_j^2 - \sum V_i^2/\sigma_j^2, \\
\log \Lambda_i &= \sum_i (a_{ij}V_j/\sigma_i^2) - \frac{1}{2} \sum_j (a_{ij}^2/\sigma_j^2).
\end{aligned}
$$

The first term on the right side is the scaled correlation coefficient; the second term is a bias term correction which corresponds to the power content of the reference waveform.

Thus, the correlation coefficient is an optimum system if probabilities and weights of the system are equal, as might be encountered in actual practice, and if the bias term for waveform power is considered.

IVAN FLORES
Remington Rand Univac
South Norwalk, Conn.

* Received by the PGEC, February 24, 1958.

[1] C. K. Chow, 1957 IRE WESCON CONVENTION RECORD, pt. 4, pp. 121–129. Also, IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 247–254; December, 1957.

A Note on the Reduction of Switching Functions*

The purpose of this note is to show how to reduce switching functions of high numbers of variables while taking advantage of redundancies.

Any switching function of n variables can be expressed in canonical form as

$$f(x_1, x_2, \cdots, x_n) = \sum_{i=0}^{M} g_i p_i \quad (M = 2^n - 1) \quad (1)$$

where the p_i's are the complete products (minterms) and the g_i's are binary coefficients peculiar to a given switching function. Eq. (1) can also be written in terms of complete products q_j of any group of four of the n variables and complete products v_k of the remaining $n-4$ variables in the form

$$
\begin{aligned}
f(x_1, x_2, \cdots, x_n) &= \sum_{j=0}^{15} \sum_{k=0}^{R} r_{kj} v_k q_j \\
&\qquad (R = 2^{n-4} - 1) \quad (2) \\
&= \sum_{j=0}^{15} \theta_j(k) q_j \\
&\qquad (k = 0, 1, 2, \cdots, R). \quad (3)
\end{aligned}
$$

In the language of the hypercube geometrical model[1] a switching function can be reduced if and only if there are at least two adjacent vertices on the associated hypercube. If there are adjacent vertices on the hypercube for n variables, these must appear either as adjacent vertices on the hypercube for the four variables, or on the hypercube for the remaining variables, but not on both of these hypercubes. A cell which is not a basic cell of the four-dimensional hypercube is not used in forming basic cells of the n-dimensional hypercube. Cells which are basic cells for the four-dimensional hypercube may *separately or collectively* form portions of basic cells for the n-dimensional hypercube.[2] When basic cells of the four-dimensional hypercube can be combined in basic cells of the n-dimensional hypercube, such combinations appear as redundancies on the Veitch-Karnaugh map in the second step of the reduction process. This argument gives the essential basis for the reduction scheme.

In brief, a higher order switching function may be reduced by working successively with variables in groups of four. The reason for choosing four, rather than any other equally valid integer, is that sets of four variables are easily reduced by means of a Veitch-Karnaugh map. The reason for applying the map method of reduction is that it places in evidence the role of redundancies (don't-care conditions).

The method used suggested is now applied, as an example, to the determination

* Received by the PGEC, February 27, 1958.

[1] R. H. Urbano and R. K. Mueller, "A topological method for the determination of the minimal form of a Boolean function," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-5, pp. 126–132; September, 1956.

B. Harris, "An algorithm for determining minimal representations of a logic function," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-6, pp. 103–108; June, 1957.

[2] There are several sets of basic cells from the four-dimensional hypercube, and cells from different sets may be combined, as has been done in obtaining (5) from the map in Fig. 2.

TABLE I
CARRY DIGIT FOR BINARY CODED-DECIMAL ADDER

B \ A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	*d*	*d*	*d*	*d*	*d*	*d*
1	0	0	0	0	0	0	0	0	0	1	*d*	*d*	*d*	*d*	*d*	*d*
2	0	0	0	0	0	0	0	0	1	1	*d*	*d*	*d*	*d*	*d*	*d*
3	0	0	0	0	0	0	0	1	1	1	*d*	*d*	*d*	*d*	*d*	*d*
4	0	0	0	0	0	0	1	1	1	1	*d*	*d*	*d*	*d*	*d*	*d*
5	0	0	0	0	0	1	1	1	1	1	*d*	*d*	*d*	*d*	*d*	*d*
6	0	0	0	0	1	1	1	1	1	1	*d*	*d*	*d*	*d*	*d*	*d*
7	0	0	0	1	1	1	1	1	1	1	*d*	*d*	*d*	*d*	*d*	*d*
8	0	0	1	1	1	1	1	1	1	1	*d*	*d*	*d*	*d*	*d*	*d*
9	0	1	1	1	1	1	1	1	1	1	*d*	*d*	*d*	*d*	*d*	*d*
10	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*	*d*
11																
12																
13								all *d*								
14																
15																

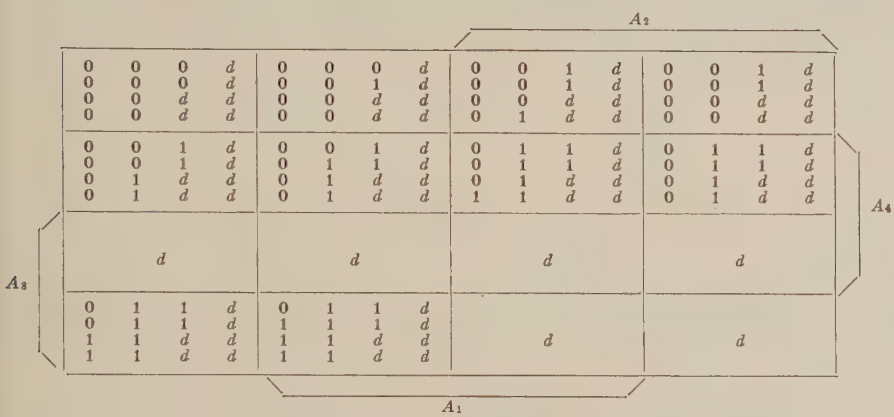


Fig. 1—First reduction with A variables.

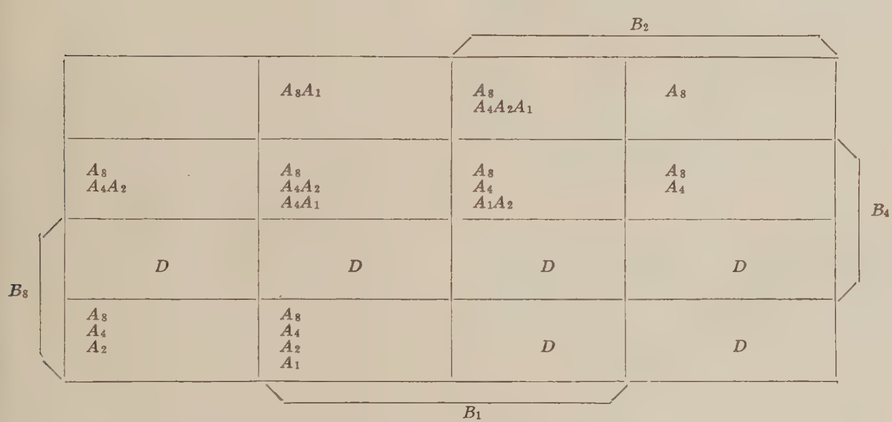


Fig. 2—Second reduction with B variables.

of a switching function for the carry digit in a binary coded decimal adder with inputs A_8, A_4, A_2, A_1, B_8, B_4, B_2, B_1. Table I shows the addition table for producing a carry.

The first reduction will be performed on the set of A variables, with the corresponding binary coefficients of the complete products of the B variables shown in groups of four on the map in Fig. 1. The reduction is carried out by dealing first with the initial element in all boxes of the map, then with the second element, and so on.

The resulting switching function is

$$\begin{aligned} C = {} & v_1A_8A_1 + v_2A_8 + v_3(A_8 + A_4A_2A_1) \\ & + v_4(A_8 + A_4A_2) + v_5(A_8 + A_4A_2 + A_4A_1) \\ & + v_6(A_8 + A_4) + v_7(A_8 + A_4 + A_1A_2) \\ & + v_8(A_8 + A_4 + A_2) + v_9(A_8 + A_4 + A_2 + A_1) \quad (4) \end{aligned}$$

where the v_i are the complete products of the B variables.

Next, reduction is carried out with respect to the B variables as shown in the map in Fig. 2. The resulting switching function for the carry digit is

$$\begin{aligned} C = {} & A_8(B_8 + B_2 + B_4) + A_4(B_4B_2 + B_8) \\ & + A_2(B_8) + A_1(B_8B_1) + A_8A_1(B_1) \\ & + A_4A_2(B_4) + A_2A_1(B_4B_2B_1) + A_4A_1(B_4B_1) \\ & + A_4A_2A_1(B_2B_1). \qquad (5) \end{aligned}$$

In obtaining (4) and (5) the redundancies have been used.

JOHN N. WARFIELD
Purdue University
Lafayette, Ind.

Logical Design Using the Stroke Function*

INTRODUCTION

Recently there has appeared an article[1] on a transistor circuit similar to the type shown in Fig. 1. The operation of the circuit is such that if any of the inputs are positive the transistor is turned off and the output is negative. If all of the inputs are negative the transistor is turned on and the output is positive. Positive and negative in this instance refer to potential levels of ground and $-E_3$, respectively. A "polarity truth table" for such a circuit having two inputs is

A_1	+	−	+	−
A_2	+	+	−	−
B	−	−	−	+

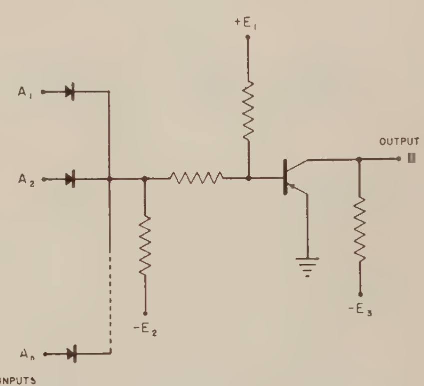


Fig. 1—Transistor gate circuit.

Using a positive logic (+ is identified with true and ONE) the circuit represents the binary NOR function, $\overline{A_1 + A_2} = B$. A negative logic (− is identified with true and ONE) results in the binary NAND function, $\overline{A_1 \cdot A_2} = B$.

SHEFFER STROKE

The Sheffer stroke is a function which has not been applied to any great extent to the design of logical circuits. The function is stated as: $A_1 | A_2$ is true if either A_1 or A_2 or both are false. An examination of the "polarity truth table" shows that the circuit in Fig. 1 can be represented by this function. Using the negative logic convention one has

0 = + = false
1 = − = true

A_1	0	1	0	1
A_2	0	0	1	1
B	1	1	1	0

$$B = A_1 | A_2.$$

The positive logic can also be used, giving

0 = − = false
1 = + = true

A_1	1	0	1	0
A_2	1	1	0	0
B	0	0	0	1

$$B = \overline{\bar{A}_1 | \bar{A}_2}.$$

* Received by the PGEC, March 14, 1958. This work was supported by the Office of Naval Research under Contract No. N7onr41906 and the Dept. of the Navy under Contract No. N600(167)45550.

[1] W. D. Rowe, "The transistor NOR circuit," 1957 IRE WESCON CONVENTION RECORD, pt. 4, pp. 231–245.

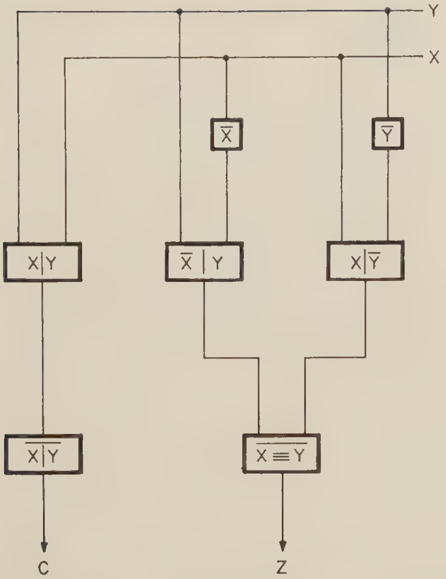


Fig. 2—Half adder based on the stroke function using circuits of the type shown in Fig. 1.

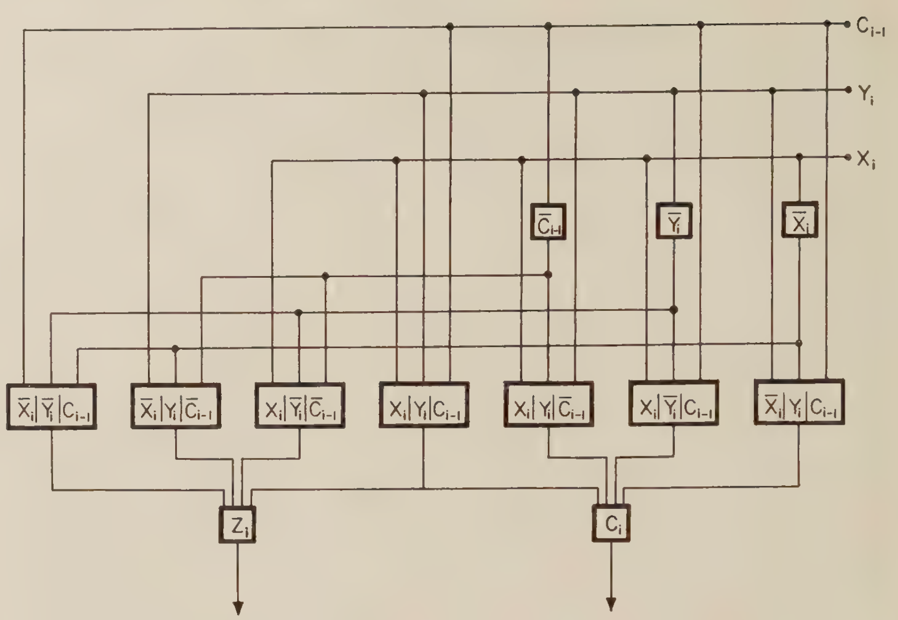


Fig. 3—Full adder based on the modified stroke function using circuits of the type shown in Fig. 1.

For convenience we choose the negative logic representation.

The relations between the stroke function and the more familiar Boolean functions are

$$A \mid A = \bar{A}, \tag{1}$$

$$\overline{A_1 \mid A_2} = A_1 \cdot A_2, \tag{2}$$

$$\bar{A}_1 \mid \bar{A}_2 = A_1 + A_2. \tag{3}$$

Two other relations, well suited for computer circuits, are

$$(\bar{A}_1 \mid \bar{A}_2) \mid (A_1 \mid A_2) = (A_1 \equiv A_2), \tag{4}$$

$$(\bar{A}_1 \mid A_2) \mid (A_1 \mid \bar{A}_2) = \overline{(A_1 \equiv A_2)}. \tag{5}$$

Using the stroke functions in (1)–(3) one can design circuits consisting only of units of the type shown in Fig. 1, which will perform all of the logical operations necessary in computers. Relations (4) and (5) are introduced since they appear in registers employing full and half adders.

Design of Circuits Using the Stroke Function

Using half and full adders as examples of the design of logical circuits involving the stroke notation and blocks consisting of the circuit shown in Fig. 1, one proceeds as follows.

The truth table for a half adder is shown below, where X and Y are the binary digits to be added, and Z and C are the sum and carry, respectively.

X	0	1	0	1
Y	0	0	1	1
Z	0	1	1	0
C	0	0	0	1

It is readily apparent that the sum, Z, can be formed by

$$Z = (\bar{X} \mid Y) \mid (X \mid \bar{Y}),$$

and the carry is just

$$C = \overline{X \mid Y}.$$

The half adder shown in Fig. 2 is constructed of blocks, each of which represents a circuit of the type shown in Fig. 1. The function of each block is inscribed in the block.

A full adder can be constructed in the manner of using two half adders and an OR gate but an economy of circuitry can be effected by working directly with the results of a truth table.

X_i	0	1	0	1	0	1	0	1
Y_i	0	0	1	1	0	0	1	1
C_{i-1}	0	0	0	0	1	1	1	1
Z_i	0	1	1	0	1	0	0	1
C_i	0	0	0	1	0	1	1	1

In the above table X_i and Y_i are the ith digits to be added, C_{i-1} is the carry from the previous stage, Z_i is the sum, and C_i is the carry. No great effort is required to show that

$$\begin{aligned} Z_i &= \overline{[C_{i-1} \equiv \overline{(X_i \equiv Y_i)}]} \\ &= \overline{[X_i \equiv \overline{(C_{i-1} \equiv Y_i)}]} \\ &= \overline{[Y_i \equiv \overline{(C_{i-1} \equiv X_i)}]} \end{aligned}$$

and

$$C_i = [\bar{C}_{i-1} \mid \overline{(X_i \mid Y_i)}] \mid [C_{i-1} \mid (\bar{X}_i \mid \bar{Y}_i)].$$

The design of the full adder can now proceed on the basis of forming the Z_i and C_i functions from (1) through (5) and then following a functional design of the type exhibited in Fig. 2.

Modified Stroke Function

Restricting the circuit in Fig. 1 to only two inputs is a waste of the circuit's capability. True, there is a practical upper limit to the number of inputs determined by the finite back resistance of the diodes but this limit is certainly greater than two. To describe the action of a circuit having n inputs we introduce a modified stroke function and define it by stating that: $A_1 \mid A_2 \mid \cdots \mid A_n$ is true if $A_1, A_2, \cdots, A_{n-1}$, or A_n is false (using the inclusive OR). Thus in negative logic a truth table for three variables would be

A_1	0	1	0	1	0	1	0	1
A_2	0	0	1	1	0	0	1	1
A_3	0	0	0	0	1	1	1	1
B	1	1	1	1	1	1	1	0

$$B = A_1 \mid A_2 \mid A_3.$$

Returning to the example of the full adder we see that we desire to construct Z_i and C_i from combinations of X_i, Y_i, and C_{i-1}, taking all three at a time. Examination of the truth table above shows that $A_1|A_2|A_3$ is "0" only when A_1, A_2, and A_3 are all "1's," therefore it is possible to obtain a "0" at any position by use of the NOT functions of A_1, A_2, and A_3. A list of these relations is given below.

$A_1 \mid A_2 \mid A_3$	1	1	1	1	1	1	1	0
$\bar{A}_1 \mid A_2 \mid A_3$	1	1	1	1	1	1	0	1
$A_1 \mid \bar{A}_2 \mid A_3$	1	1	1	1	1	0	1	1
$\bar{A}_1 \mid \bar{A}_2 \mid A_3$	1	1	1	1	0	1	1	1
$A_1 \mid A_2 \mid \bar{A}_3$	1	1	1	0	1	1	1	1
$\bar{A}_1 \mid A_2 \mid \bar{A}_3$	1	1	0	1	1	1	1	1
$A_1 \mid \bar{A}_2 \mid \bar{A}_3$	1	0	1	1	1	1	1	1
$\bar{A}_1 \mid \bar{A}_2 \mid \bar{A}_3$	0	1	1	1	1	1	1	1

Referring to the above list and the truth table for the full adder it is readily apparent that

$$Z_i = (X_i \mid Y_i \mid C_{i-1}) \mid (\bar{X}_i \mid \bar{Y}_i \mid C_{i-1}) \mid \cdot (\bar{X}_i \mid Y_i \mid \bar{C}_{i-1}) \mid (X_i \mid \bar{Y}_i \mid \bar{C}_{i-1}),$$

and

$$C_i = (X_i \mid Y_i \mid C_{i-1}) \mid (\bar{X}_i \mid Y_i \mid C_{i-1}) \mid \cdot (X_i \mid \bar{Y}_i \mid C_{i-1}) \mid (X_i \mid Y_i \mid \bar{C}_{i-1}).$$

A block diagram of a full adder based on circuits of Fig. 1 and the above relations for Z_i and C_i is shown in Fig. 3.

Various methods of notation can be used to designate complex stroke functions other than the usual set of brackets, parentheses, etc. In fact, it is possible to designate all

possible functions using only the stroke notation since the NOT function has an equivalent stroke notation. This is done by replacing the brackets with multiple strokes and the NOT function with its equivalent stroke function. As an example, consider the sum and carry functions mentioned above. They could be written

$$Z_i = X_i \| Y_i \| C_{i-1} \||\, X_i \,|\, X_i \| Y_i \,|\, Y_i \| C_{i-1} \||\, X_i \,| \cdot X_i \| Y_i \| C_{i-1} \,|\, C_{i-1} \||\, X_i \| Y_i \,|\, Y_i \| C_{i-1} \,|\, C_{i-1},$$

$$C_i = X_i \| Y_i \| C_{i-1} \||\, X_i \,|\, X_i \| Y_i \| C_{i-1} \||\, X_i \| Y_i \,| \cdot Y_i \| C_{i-1} \||\, X_i \| Y_i \| C_{i-1} \,|\, C_{i-1}.$$

Obviously this is a matter of one's own taste and the authors have found that their preference is to use brackets, the NOT function, and to omit the stroke in the lowest level of the logic so that the sum and carry would appear as

$$Z_i = X_iY_iC_{i-1} \,|\, \bar{X}_i\bar{Y}_iC_{i-1} \,|\, \bar{X}_iY_i\bar{C}_{i-1} \,|\, X_i\bar{Y}_i\bar{C}_{i-1},$$

$$C_i = X_iY_iC_{i-1} \,|\, \bar{X}_iY_iC_{i-1} \,|\, X_i\bar{Y}_iC_{i-1} \,|\, X_iY_i\bar{C}_{i-1}.$$

Higher order functions would of course require the use of brackets, parentheses, etc. No loss in meaning or understanding results from this "simplified" notation but the complex notation using multiple strokes does provide a convenient means for determining the number of transistors in the circuit. For example, in the Z_i relation there are three different single stroke relations, four different double stroke relations, and one triple stroke relation making a total of eight transistors required to generate Z_i. Only twelve transistors are required to generate both Z_i and C_i because four of the intermediate functions are common to both. This also is evident when the two functions are compared with each other.

Conclusions

A symbolic notation has been developed which fits the transistor gate circuit shown in Fig. 1 to the Sheffer stroke function. An extension of the stroke function has been developed to aid in the application of the logic to complex circuits. The two examples shown are relatively simple circuits but the extension to more complex circuits is a simple matter. It is felt that a considerable simplification in the design of computer circuitry can be achieved if the elemental circuit blocks are all of the same form, allowing one to express all logical operations in the form of the stroke function.

N. T. Grisamore
L. S. Rotolo
G. U. Uyehara
George Washington Univ.
Washington, D. C.

Contributors

D. D. Aufenkamp, for a biography see page 205 of the September, 1957, issue of these Transactions.

Walter F. Bauer (A'53) was born in Wyandotte, Mich., on March 21, 1924. He received the B.S., M.S., and Ph.D. degrees from the University of Michigan, Ann Arbor. He was an Air Force weather officer during World War II, and research engineer at the Willow Run Research Center, University of Michigan from 1951 to 1954.

He has been with Ramo-Wooldridge Corporation since 1954, and currently directs the Computation and Data Reduction Center, Space Technology Laboratories. His main interests are system simulation, computer systems, numerical analysis, data reduction, and programming. He has been directly involved with missile simulation on digital computers since 1951 with the SEAC computer at NBS and with the MIDAC computer at the University of Michigan. He has written several papers on computer systems and simulation.

Dr. Bauer is a member of the Association for Computing Machinery and the American Mathematical Society.

J. V. Blankenbaker (A'54) was born on December 24, 1929, in Apache, Okla. Following two years as an electronic technician in the U. S. Navy, he attended Oregon State College, Corvallis, Ore., where he received the B.S. degree in 1952 in physics and mathematics. In 1954, he received the M.S. degree in physics from the University of California at Los Angeles.

From 1952 to 1956, he was engaged in the design of data processors and special purpose computers at Hughes Aircraft Company. During the summer of 1957, he did computer design studies at the Ramo-Wooldridge Corporation, Los Angeles, Calif. Since 1956, he has been working toward the Electrical Engineer degree at the Massachusetts Institute of Technology.

Mr. Blankenbaker is a member of Phi Kappa Phi and the ACM.

Edward L. Braun (SM'55) was born on February 18, 1925 in Pittsburgh, Pa. He received the B.A. degree in physics in 1947 and the M.S. degree in engineering in 1954 from the University of California at Los Angeles. While in the U. S. Navy, he received extensive education in electronics, principally in radio and radar.

His professional experience and first interest in digital computers began at Northrop Aircraft, Inc., where he stayed from 1947 to 1952. From 1952 to 1955 he was with Librascope, Inc., and the following year he was with Lockheed's Missile Systems Division. He was subsequently employed by Hughes Aircraft Company and Litton Industries, Beverly Hills, Calif. Now he is director of engineering of Genesys Corporation (a subsidiary of Chance-Vought Aircraft Company, Inc.), Los Angeles, Calif.

Mr. Braun has organized and headed a number of digital computer development programs. He has designed digital computers for military computation and control systems, and for engineering and scientific computation. While at the Lockheed Missile Systems Division, he also served for a time as head of the Digital Computing Center.

Mr. Braun is a member of the ACM, and served a term as membership chairman of the Southern California section of the ACM.

William Comley was born in Indianapolis, Ind., on September 14, 1921. After completing the electronics curriculum at Los Angeles City College, Calif., he served as a communications officer with the U. S. Air Force for three years and was for the greater part of this time in charge of a number of high-powered ground transmitting stations.

In 1947, Mr. Comley joined the engineering staff of Lear, Inc. of California as a design engineer in aircraft radio. He later served as project engineer for that company's line of phonograph pickups and wire recorder heads.

In 1950, Mr. Comley joined Douglas Aircraft Company, where he is currently engaged in the design of analog computing devices and auxiliary equipment for existing facilities.

Robert G. Gillespie was born May 18, 1935 in Tacoma, Wash. He received the B.A. degree in mathematics from Reed College, Portland, Ore., in 1955.

From 1955 to 1956, he was at Convair Aircraft Company, San Diego, Calif., in the Digital Computing Laboratory. From 1957 to 1958, he was at the Astronautics Division of Convair, where he worked on the digital simulation of missile guidance systems. He now is a member of the technical staff of the Pilotless Aircraft Division, Boeing Airplane Company, Seattle, Wash., where he is engaged in work on systems evaluation and simulation.

He is a member of the American Mathematical Society, the Association for Computing Machinery, and the Society for Industrial and Applied Mathematics.

William F. Gunning (S'41–A'43–M'55) was born on August 11, 1916, in New York, N. Y. He received the A.B. degree in physics from the University of California at Los Angeles in 1941. His professional experience started at Douglas Aircraft Company's Santa Monica Electrical Research Laboratory, where he worked on flight and static test instrumentation from 1941 to 1948. He transferred to Project Rand and stayed with this organization when it became the Rand Corporation in 1948. He was in charge of rebuilding an early model REAC analog computer, adding chopper stabilizer amplifiers, and the first centralized removable program plugboard and operator's console to be used on a large analog computer installation. After a tour on part-time loan to the group constructing the SWAC at U.C.L.A., he became project engineer on Rand's Johnniac digital computer which is a member of the Princeton I.A.S. family of machines.

From 1953 to 1956, he was with International Telemeter Corporation working on problems associated with pay-as-you-see TV and magnetic core storage systems.

He joined the Systems Division of Beckman Instruments in 1956 where he is Chief Project Engineer working on data logging and data acquisition equipment for industrial application.

Mr. Gunning has served on several Joint Computer Conference committees.

❖

Ladis D. Kovach (A'53) was born on November 21, 1914 in Budapest, Hungary. He received the B.S. degree in physics in 1936 and the M.S. degree in mathematics in 1948 from Case Institute of Technology, Cleveland, Ohio; the M.A. degree in education from Western Reserve University, Cleveland, Ohio, in 1940; and the Ph.D. degree in mathematics from Purdue University, Lafayette, Ind., in 1951.

From 1936 to 1948, he was associated with Picker X-Ray Corporation, American Shipbuilding Company, and Ohio Crankshaft Company as electrical designer. He was an instructor in mathematics from 1946 to 1951, first at Case and then at Purdue. Since 1951, he has been with the Douglas Aircraft Company as a design specialist.

Dr. Kovach is a member of the American Physical Society, Sigma Xi, and the American Mathematical Society.

❖

Morton W. Marcovitz (S'50–A'52–M'57) was born on September 20, 1927, in Philadelphia, Pa. He received the B.S.E.E. degree from Drexel Institute of Technology in 1951, and the M.S.E.E. degree from the University of Pennsylvania in 1957.

He has been associated with the research activity of the Burroughs Corporation since 1951, working on applications of digital computer techniques to data processing systems.

Mr. Marcovitz is a member of Eta Kappa Nu, Tau Beta Pi, AIEE, ACM, and RESA.

❖

Douglas B. Netherwood (M'56) was born in Dallas, Texas, on May 8, 1920. He received the B.S. degree from the U. S. Military Academy in 1943 and the M.S. degree in electronics from the Air Force Institute of Technology in 1957. He served in India and Burma during World War II and was Advisor on Research and Development to the Nationalist Government of China from 1946 to 1947. He was a Communications Officer in the Strategic Air Command from 1947 to 1955. At present he is Chief of the Electron Tube and Circuit Research Section, Electronic Components Laboratory, Wright Air Development Center, Ohio. His field of special interest is the simplification of electronic components and networks.

Major Netherwood is a member of the ACM and Association for Symbolic Logic.

❖

Palakodety Venkata Rao was born August 18, 1922. He received the Master's degree in physics from Andhra University, Waltair, India, in 1943, and then studied electrical engineering at the Indian Institute of Science. He later worked as a design electrical engineer with the India Electrical Works in Calcutta, and since 1948, has been a member of the faculty of the Indian Institute of Science at Bangalore. He will shortly complete his work for the D.Sc. degree in electrical engineering at Andhra University.

Mr. Rao is an associate member of the AIEE.

❖

Joseph H. Rawlings was born in Omaha, Neb. on April 16, 1924. He received the B.S. degree from Western Reserve University, Cleveland, O., in 1952.

He has been with the National Advisory Committee for Aeronautics since 1946 and is now head of Computer Engineering at the Lewis Flight Propulsion Laboratory, Cleveland.

❖

Eric Seif (S'52–A'53) was born on September 25, 1925, in Vienna, Austria. He received the B.S. degree in electrical engineering from Drexel Institute of Technology, Philadelphia, Pa., in 1952 and the M.S. degree from the University of Pennsylvania, Philadelphia, Pa., in 1957.

Since 1952 he has been with Burroughs Corporation Research Center, Paoli, Pa.

Mr. Seif is a member of Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, and RESA.

❖

Alfred K. Susskind (S'47–A'50–M'56) was born on October 4, 1923, in Coblenz, Germany. He received a B.E.E. degree from the Polytechnic Institute of Brooklyn in 1948, and the S.M. degree from M.I.T., Cambridge, Mass., in 1950.

From 1948 to 1950, he was a research assistant in the Digital Computer Laboratory at M.I.T., working on the electronic design for the Whirlwind Computer. From 1950 to 1955, he was a research engineer in the Servomechanisms Laboratory, working on numerical control of machine-tools.

Since 1955, he has been an assistant professor in the Department of Electrical Engineering at M.I.T., giving courses in digital data processing engineering, and doing research in airborne data-processing problems.

Mr. Susskind is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.

❖

L. R. Turner was born on August 19, 1910, in Williamsport, Pa. He received the B.S. degree in physics in 1931 and the M.S. degree in 1932 from Pennsylvania State University, University Park.

He has been with the National Advisory Committee for Aeronautics since 1938, working in engineering until 1949 and then in applied mathematics. He is Chief of the Mechanized Computation and Analysis Branch at Lewis Flight Propulsion Laboratory, Cleveland, Ohio.

He is a member of the ACM, Pi Mu Epsilon, and Sigma Rho Sigma.

JOINT COMPUTER COMMITTEE

SENEWS

SCIENCE EDUCATION SUBCOMMITTEE NEWSLETTER

Vol. 1, No. 1 July, 1958

What Is *SENEWS*?

A newsletter addressed to computer-oriented members of IRE, ACM, and AIEE to help them promote computer interest and knowledge among high-school age students.

Who Is Responsible?

The Science Education Subcommittee under sponsorship of the Joint Computer Committee. C. W. Farr of Lincoln Laboratory, M.I.T., is Chairman of the Subcommittee and Acting Editor of *SENEWS*.

Why Is It?

Computers are here to stay. High-school students are already showing a budding curiosity about computers. Professional computer people are showing wide interest in helping the kids to get started. All over the country computer educational activity is springing up. A rallying point is needed for sharing ideas and results.

Send Us A Computer Story

. . . of your local high-school activity. If there is no news story, guess you'll have to go out and *make* some news. Let's challenge the youngsters to reach their potential strength in the data processing world of tomorrow. Watch *SENEWS* for ideas, and send us your success stories!

Write to: C. W. Farr, JCC Science Education Subcommittee, M.I.T. Lincoln Laboratory, Lexington 73, Mass.

Have Design, Will Build

A high-school junior and perennial regional science fair winner from Cedar Falls, Iowa, named David Ecklein, has designed and is building a thirty-five hundred vacuum tube digital computer, using surplus market tubes and sockets and adapting office shelving for use as chassis and cabinet material. He has designed the mathematical logic and circuitry with essentially no assistance other than normal instruction in high-school science, what he has been able to learn from books, and his own experiments. Part of the circuitry has been assembled, with the help of a high-school sophomore friend whom he, in turn, has helped on another science project.

The point of the story is that the ability and motivation for a budding computer engineer can spring up in the middle of Iowa or in the suburbs of Boston or Los Angeles, for example. There is one big difference: in the coastal industrialized areas, computers and computer engineers abound, and a youngster finds it is easy to obtain local adult professional help in the pursuit of his project.

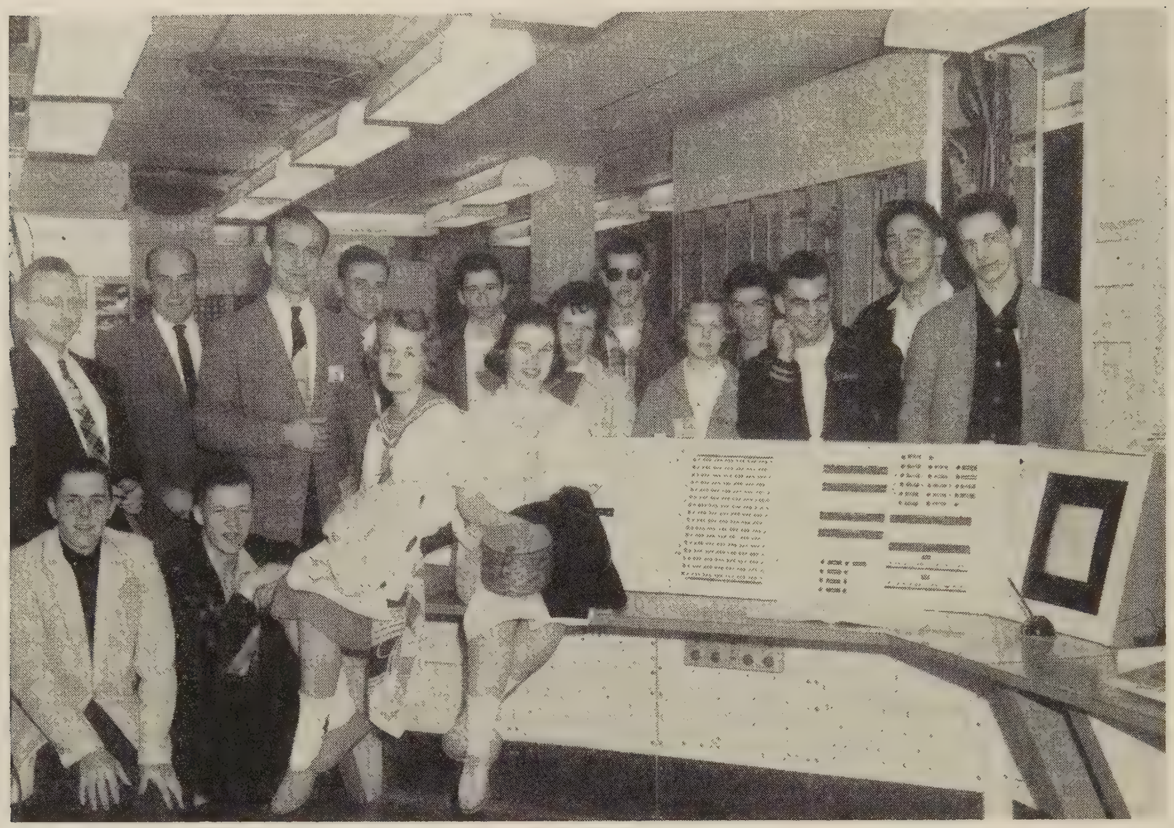
The JCC Science Education Subcommittee heard of this boy by way of Washington, D. C. (the President's Committee on Scientists and Engineers), and is proud to have brought him to the attention of JCC members who have set the wheels in motion to give him well deserved counsel and encouragement. (*Ed. note:* David has sold over 500 Bibles to raise money for his science project. At last report he was flying to New York on a Saturday–Tuesday visit as guest of IBM.)

Low-Budget High-School Computer

In Buffalo, N. Y., Aaron Buchman, mathematics instructor at the Hutchinson Central Technical High School, has been conducting a course in computer programming and coding open to college preparatory seniors. Under his direction, members of the Tech Math Club constructed a tape activated automatic digital computer for use in the course.

The one-semester course consists of theory and laboratory work. After 30 classes in theory, the students get two laboratory and three theory periods per week. Included in the theory are number systems, drill in binary system arithmetic operations, elements of Boolean algebra, and computer logic.

The computer is an electromagnetic relay machine which operates in the binary number system. Tapes are



A. Vanderburgh, R. Enticknap, N. Levy, and R. Mayer (back row, left to right) conduct a visit of Concord High School students to M.I.T. Lincoln Laboratory's experimental computer, TX-2.

punched out of pressboard nine inches wide, and carry ten channels. Each of the six registers of the storage section holds a six-digit binary number. Panel lights indicate the contents of each register at all times. With two operator registers and one output register, the arithmetic unit performs direct addition, subtraction, and multiplication. Division is accomplished by a taped routine.

Before completing the course, each pupil must program and code an approved program which requires branch points, cycling routines, a minimum number of orders, and which must run without trouble on the computer.

Among the problems successfully programmed and now a part of the library of routines are those to:

1) Find the total surface area of a rectangular solid'
2) Find the average of two signed numbers,
3) Evaluate a second degree polynomial using successive integer values of the variable,
4) Solve simultaneous linear equations,
5) Evaluate a two-rowed determinant,
6) Evaluate various formulas from mathematics and physics for sets of values of the variables involved by having the computer order the machine to take a new value of one of the parameters with each pass of the tape.

A description of the machine and the mathematics course in which it is used was presented by Mr. Buchman at the ACM meeting in August, 1956. *Mathematics Teacher* carried the story in March, 1958. The course now includes three weeks of IBM 650 programming.

Here, a teacher's initiative and a forward-looking school system have combined to challenge the abilities of bright high-school students with a solid introduction to the technological revolution taking place within their time.

Students Study

Concord (Mass.) High School's Mathematics Department experimented in the 1958 spring term with a six-week curriculum unit on computers. Fifteen students participated in the unit, which was designed by Norton Levy of the Concord High School Mathematics Department, assisted by M.I.T. Lincoln Laboratory staff members, who acted as consultants, arranged computer facilities tours, and furnished a demonstration-lecture team for a major part of the course. Lectures included number systems, computer principles (slanted away from electronic circuitry), simple programming theory, study of actual coding of two programs for the IBM 704 computer, a follow-up visit to a computer facility while the programs were being run, and computer applica-

tions. Course evaluation was by written examination. This curricular experiment, guided by a teacher-engineer team, was sponsored by Ford Foundation's School and University Program for Research and Development, under grant to Harvard's School of Education.

An interesting sidelight of this story is the "cardboard computer," a mechanical model made from cardboard and common pins by R. P. Mayer in one evening and used in a high-school lecture-demonstration. This is an outgrowth of the effort described by Mayer in his paper, "A Proposal for Training Youngsters in Digital Computing Techniques," presented to the ACM in Los Angeles in August, 1956. The "cardboard computer" was a binary unit $8\frac{1}{2}$ by 11 inches, comprising a manually activated pulse input capable of producing either a "zero" or a "one" pulse, a counter, control element selecting between two one-bit storage registers, and a one-bit adder.

So Do Teachers

A high-school teacher computer course consisting of seven lecture and workshop meetings is described in "Introduction to Arithmetic Computers," a report to the Hughes Aircraft Company by ten participating (high-school science and mathematics) teachers, Los Angeles School-Industry Science Program, December, 1956. (Copies are available from Dr. L. C. Van Atta, Head, Technical Information and Education, Hughes Aircraft Company, Culver City, Calif.) The report is intended for use as a syllabus for a teachers' 40-hour workship. The lectures reviewed computer history, binary arithmetic, elementary computer logic, and included visits to a mechanical differential analyzer, to the Hughes REAC (analog) and IBM 650 (digital), and to the Bureau of Standards SWAC (digital) computer facilities. One lecture was devoted to SWAC programming; homework called for completion of the coding of a fifth degree polynomial problem (appendix to the report contains detailed flow diagram and coded program).

This computer activity is only one phase of a well-organized community program[1] in which the school system employs a full-time school-industry coordinator who works with a Council. The Council's board of directors includes representatives of secondary schools, universities, industry, and the professional societies; the Council publishes a newsletter, organizes industrial demonstration-lecture teams for high schools, coordinates an industrial summer work program for high-school teachers, and in July, 1957 participated in an industry-education conference financed by Hughes Aircraft Company and held at the Lake Arrowhead facilities of the University of California. For information write to J. H. Cooper, Executive Secretary, Southern California Industry-Education Council, 954 West 37th Street, Los Angeles 7, Calif.

[1] L. C. Van Atta, "The Los Angeles school-industry science program," *Phys. Today*, vol. 10; September, 1957.

Did You Hear . . .

. . . the one about the computer programmer who couldn't balance his checking account. He knew that $177+1=200$; but the bank took a narrower view.

IRE, ACM, AIEE

Richard W. Melville (Computer Development Group, Stanford Research Institute), is Chairman of the PGEC Student Relations Committee, and IRE representative of the *SENEWS* Editorial Board. The original Chairman of the JCC Science Education Subcommittee, he was appointed when its charter was a nebula of swirling hot gases; he explored and charted the unknown and has been a valued consultant in establishing the present subcommittee policy and program and in bringing forth this newsletter.

Professor George E. Forsythe (Mathematics Department, Stanford University) is Chairman of the ACM Secondary School Educational Committee (actually ACM has two other education committees, University-Education and Industry-Education). Professor Forsythe is ACM representative on the *SENEWS* Editorial Board and has furnished valued criticism and encouragement. Those interested should write him for his Bibliography on High-School Mathematics Education, which includes ten books on digital computers. Professor Forsythe has earnestly questioned our editorial policy of confining *SENEWS* to *computers*, a question that has probably arisen also in the minds of many of our readers.

While it is true that *computer engineers* as individuals can be broadly helpful in high-school science education, the JCC Science Education Subcommittee finds many other groups at work in the field and has focused on the task of providing "a newsletter addressed to computer oriented members of IRE, ACM, and AIEE, to help them promote computer interest and knowledge among high-school age students."

Dr. Morris Rubinoff (Government and Industrial Division, Philco) is Chairman of the AIEE Computing Devices Committee, and has given the JCC Science Education Subcommittee the benefit of his views on our newsletter program. In the temporary overseas absence of Dr. Rubinoff we are grateful to Vice-Chairman R. A. Imm and others in AIEE for counsel and criticism in launching *SENEWS*.

Book and Film

"Where can I find publications on computers understandable to beginners?" You have probably been asked this question. Here is a short-short list of suggestions:

"Faster Than Thought," edited by Bowden, pub-

lished in 1953 by Pitman, London (available in U. S. at \$9.50 and worth it). Subtitled "A Symposium on Digital Computing Machines," it covers history and theory, surveys British and American computers, and reviews computer applications. The book is thorough and is written with "sparkle" (*e.g.*, ". . . an expert is sometimes defined as one who solves minor problems and avoids small errors as he sweeps forward to the 'Grand Fallacy'.")

"The Electronic Brain and What It Can Do," by Gorn, Manheimer, and Brandwein, published in 1956 by Science Research Associates (paperback, 50 cents). We have shown this to a range of people from high-school students and teachers to physicists, and they all liked it. It uses several card tricks to demonstrate yes-no logic.

"Making Electrons Count" is a 16-mm sound film telling the story of programming for the Whirlwind I digital computer; old (1954), but good for beginning audiences. This half-hour color film is available for mailing charges from M.I.T. Computation Center.

Feedback

If you have read this far, you are identified as a person interested in our program. How about a story of high-school computer activity from your locality, or a word of constructive criticism, or just tell us you liked our newsletter, or just tell us!

PGEC News

Patents Relating to Computers

Is there enough interest among Transactions readers to warrant publishing something in each issue about computer patents? The simplest and easiest thing to do is to publish a list giving number, date, inventor's name, assignee, and title. Many of these would be from classification 235-61. More elaborate information might include an abstract of the disclosure. Readers interested should let the editor know of their interest. They should also realize that some help from one or more of them will have to be solicited if a decision is made to proceed with the publishing of patent information.

New Affiliate

The Institution of Electrical Engineers of Great Britain has been approved as an affiliate of the PGEC.

Dr. William G. Fuller Memorial Award

The Professional Group on Component Parts has announced that several awards of \$250 will be made to seniors and graduate students for outstanding papers on the subject of component parts submitted by December 31, 1958, to any IRE publication or presented at any national meeting or symposium sponsored or supported by the IRE. Interested students are urged to consult their IRE Faculty Advisor.

Summary of PGEC Administrative Committee Meeting of March 26, 1958

The final meeting of the 1957–1958 Administrative Committee was held at IRE Headquarters, New York, N. Y. on March 26, 1958, during the IRE National Convention. A total of 15 members were present or represented by proxy.

New officers were elected for the year starting April 1, 1958:

Chairman—Willis H. Ware, Santa Monica, Calif.
Vice-Chairman—Richard O. Endres, Philadelphia, Pa.

Five new Administrative Committee members were elected for a three-year term, starting April 1, 1958:

Walter L. Anderson, Arlington, Va.
Arnold A. Cohen, St. Paul, Minn.
Daniel Haagens, Long Island City, N.Y.
Frank B. Heart, Cambridge, Mass.
Keith W. Uncapher, Santa Monica, Calif.

The First International Conference on Information Processing, as it is officially called, is now definitely set under UNESCO sponsorship for June 15–20, 1959, according to I. L. Auerbach's report. The place will be either Paris or Rome, to be decided this summer, and the method of selecting papers is being worked out.

The conference subjects are listed under these headings:

Methods of digital computing,
Logical design and common symbolic language for machines,
Automatic translation of languages,
Collection, storage, and retrieval of information,
Pattern recognition and machine learning.

A lively discussion followed the report of the Bibliography Committee. Committee Chairman L. F. Jones presented several ways in which the series of abstracts in the Transactions, formerly prepared under the editorship of H. D. Huskey, could be continued. A distinction must be made between the regular publication of abstracts or reviews of current literature and the occasional publication of a bibliography of computer literature up to a certain date, including careful classification and cross-indexing of the material. It was concluded that the abstracts were the more urgent at this time to avoid too long a gap after the termination of Huskey's series. A bibliography, however, presents an equally important task, as well as a more difficult one, and the Committee will work on it concurrently.

The interest at the meeting indicates that the preparation of a good bibliography and continuation of the abstract series could be a most important service for the PGEC to give its members. In recognition of this, the Bibliography Committee was elevated to the rank of a permanent committee.

At the conclusion of another successful year for the PGEC, I wish to thank members of the Administrative Committee and the various committee chairmen for their efforts and cooperation. The PGEC is the largest Professional Group in the IRE and it is still growing. This reflects not only the continuing growth and importance of the computer field, but also the contributions of the active members which have made the PGEC an important institution in the technical field.

In particular, I would like to mention Ike Auerbach who initiated the organization of the First International Conference on Information Processing (which somebody, following a venerable international custom, will surely label ICIP some day). Ike is now PGEC representative and chairman of the NJCC Committee which organizes the United States' end of the conference. He has put in a great deal of work during the past year, contacting many organizations in this and other countries, and making a trip to Paris which culminated in the conference becoming firmly established. Ike's current unofficial title is Minister of Foreign Affairs.

I hope that this example will inspire other PGEC members to seek a field of activity in which they can make an effective and unique contribution.

W. Buchholz
Chairman

INFORMATION FOR AUTHORS

The PGEC TRANSACTIONS is published quarterly and will bear datelines of March, June, September, and December. Abstracts of papers appearing in the TRANSACTIONS will appear also in IRE PROCEEDINGS. The PGEC publication schedule requires about one month for review and correction of all accepted manuscripts. The professional IRE Editorial Staff requires an additional two months' production time from receipt of manuscripts to completion of the printed journal.

MANUSCRIPTS: Three copies of the manuscript should be submitted. They should be typewritten (original and two carbon copies), and double spaced on only one side of each sheet. References should appear as footnotes, numbered consecutively, and include in the following order the author's name (including initials), title of reference work, journal name, volume, initial and final page numbers, and date of publication. Footnotes should be listed on a separate sheet and not inserted in text. Each paper must be accompanied by two copies of a summary not more than 200 words in length. Reviewing normally will require about four weeks from receipt of manuscript.

ILLUSTRATIONS: Only original illustrations should be submitted; they will be returned if desired. Photostatic copies of originals are not acceptable, except where they are exceptionally clear, with sharp black and white contrasts. All line drawings (graphs, charts, diagrams, etc.) should be prepared on drafting cloth or white drawing paper in India ink. It is preferable that only the coordinate lines show in graphs. All lettering must be large enough to be legible when reduced 50 to 75 per cent in size. Photographs should be glossy prints. Figure numbers should be indicated on the back of each illustration. Figure numbers and captions should be listed on a separate sheet accompanying manuscript. All drawings, photographs, and other manuscript material should be not larger than $8\frac{1}{2}$ by 11 inches for ease in handling.

Please submit all manuscripts to

J. P. Nash, PGEC Editor

Missile Systems Division, Lockheed Aircraft Corp.,

3251 Hanover Street, Palo Alto, California